

Docket No.: 57454-116

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Toyohiko YOSHIDA, et al.

Serial No.:

Group Art Unit:

Filed: May 16, 2001

Examiner:

For: DATA PROCESSING APPARATUS OF HIGH SPEED PROCESS USING MEMORY
OF LOW SPEED AND LOW POWER CONSUMPTION



**CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT**

Commissioner for Patents
Washington, DC 20231

Sir:

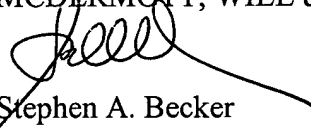
In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claim the priority of:

Japanese Patent Application No. 2000-257231,
filed August 28, 2000

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY


Stephen A. Becker
Registration No. 26,527

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 SAB:dtb
Date: May 16, 2001
Facsimile: (202) 756-8087

BEST AVAILABLE COPY

57454-116

Yoshida, et al

May 16, 2001

日本国特許庁

PATENT OFFICE
JAPANESE GOVERNMENT

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as with this Office.

出願年月日

Date of Application:

2000年 8月28日

出願番号

Application Number:

特願2000-257231

出願人

Applicant(s):

三菱電機株式会社

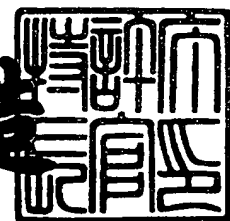
jc868 U.S. Pat. & Tm. Off.
09/855894
05/16/01

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 9月18日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3075117

【書類名】 特許願

【整理番号】 525067JP01

【提出日】 平成12年 8月28日

【あて先】 特許庁長官殿

【国際特許分類】 G11C 7/00
G06F 12/02

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社
社内

【氏名】 吉田 豊彦

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社
社内

【氏名】 山田 朗

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社
社内

【氏名】 佐藤 尚和

【特許出願人】

【識別番号】 000006013

【氏名又は名称】 三菱電機株式会社

【代理人】

【識別番号】 100064746

【弁理士】

【氏名又は名称】 深見 久郎

【選任した代理人】

【識別番号】 100085132

【弁理士】

【氏名又は名称】 森田 俊雄

【選任した代理人】

【識別番号】 100091409

【弁理士】

【氏名又は名称】 伊藤 英彦

【選任した代理人】

【識別番号】 100096781

【弁理士】

【氏名又は名称】 堀井 豊

【選任した代理人】

【識別番号】 100096792

【弁理士】

【氏名又は名称】 森下 八郎

【手数料の表示】

【予納台帳番号】 008693

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ処理装置

【特許請求の範囲】

【請求項 1】 命令が格納される命令メモリと、

データが格納されるデータメモリと、

フェッチされた命令をデコードする命令デコーダと、

前記命令メモリ、前記データメモリおよび前記命令デコーダに接続され、前記命令メモリに格納される命令をフェッチし、前記命令デコーダのデコード結果に基づいて前記データメモリにアクセスするメモリ演算部と、

前記命令デコーダのデコード結果に基づいて整数演算を行なう整数演算部とを含むデータ処理装置であって、

前記命令メモリは、複数の命令メモリバンクを含み、

前記メモリ演算部は、前記複数の命令メモリバンクから命令をフェッチする場合に、命令メモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとを発生させてパイプライン処理を行なう、データ処理装置。

【請求項 2】 前記命令メモリはさらに、連続したアドレスにアクセスした場合に、前記複数の命令メモリバンクの異なる命令メモリバンクにアクセスされるように、下位アドレスを含んだアドレスをデコードして前記複数の命令メモリバンクのチップセレクト信号を生成する第 1 のバンク選択回路を含む、請求項 1 記載のデータ処理装置。

【請求項 3】 前記命令メモリはさらに、高速命令メモリを含み、

前記メモリ演算部は、前記高速命令メモリから命令をフェッチする場合に、前記命令メモリバンクの選択に対応したパイプラインステージを発生させずに、命令の読出しに対応したパイプラインステージを発生させてパイプライン処理を行なう、請求項 1 または 2 記載のデータ処理装置。

【請求項 4】 前記データメモリは、複数のデータメモリバンクを含み、

前記メモリ演算部は、前記複数のデータメモリバンクにアクセスする場合に、データメモリバンクの選択に対応したパイプラインステージと、データのアクセ

スに対応したパイプラインステージとを発生させてパイプライン処理を行なう、請求項 1 ～ 3 のいずれかに記載のデータ処理装置。

【請求項 5】 前記データメモリはさらに、前記複数のデータメモリバンクを 2 つの異なる領域に分割するために、上位アドレスを含んだアドレスをデコードして前記複数のデータメモリバンクのチップセレクト信号を生成する第 2 のバンク選択回路を含む、請求項 4 記載のデータ処理装置。

【請求項 6】 前記第 2 のバンク選択回路は、前記 2 つの異なる領域の中の連続したアドレスにアクセスした場合に、前記複数のデータメモリバンクの異なるデータメモリバンクにアクセスされるように、下位アドレスを含んだアドレスをデコードして前記複数のデータメモリバンクのチップセレクト信号を生成する、請求項 5 記載のデータ処理装置。

【請求項 7】 前記データメモリはさらに、高速データメモリを含み、前記メモリ演算部は、前記高速データメモリにアクセスする場合に、前記データメモリバンクの選択に対応したパイプラインステージを発生させずに、データのアクセスに対応したパイプラインステージを発生させてパイプライン処理を行なう、請求項 4 ～ 6 のいずれかに記載のデータ処理装置。

【請求項 8】 前記メモリ演算部は、命令バスを介して前記命令メモリからの命令のフェッチを行ない、前記命令バスと異なるデータバスを介して前記データメモリにアクセスする、請求項 1 ～ 7 のいずれかに記載のデータ処理装置。

【請求項 9】 前記メモリ演算部は、データ入力バスを介して前記データメモリからデータを読み出し、前記データ入力バスと異なるデータ出力バスを介して前記データメモリにデータを書込む、請求項 1 ～ 8 のいずれかに記載のデータ処理装置。

【請求項 1 0】 命令が格納される命令メモリと、
データが格納されるデータメモリと、
フェッチされた命令をデコードする命令デコーダと、
複数のレジスタを有するレジスタファイルと、
前記命令メモリ、前記データメモリおよび前記命令デコーダに接続され、前記命令メモリに格納される命令をフェッチし、前記命令デコーダのデコード結果に

基づいて前記データメモリにアクセスするメモリ演算部と、

前記命令デコーダのデコード結果に基づいて整数演算を行なう整数演算部とを含むデータ処理装置であって、

前記メモリ演算部は、リピート命令を実行する場合に、前記リピート命令の直後の命令を前記レジスタファイル内の専用レジスタに保持し、該専用レジスタに保持された命令をフェッチしながらリピート命令を実行する、データ処理装置。

【請求項 1 1】 前記レジスタファイルは、プロセッサ状態ワードを含み、

前記メモリ演算部は、リピート命令を実行する場合に、最初のループにおいて前記プロセッサ状態ワード内のフラグをセットし、前記命令メモリからフェッチした前記リピート命令の直後の命令を前記専用レジスタに保持し、

2 回目のループにおいて前記プロセッサ状態ワード内の前記フラグをリセットし、前記専用レジスタに保持された命令をフェッチしながらリピート命令を実行する、請求項 1 0 記載のデータ処理装置。

【請求項 1 2】 前記メモリ演算部は、リピート命令を実行する場合に、前記リピート命令の直後の複数の命令を前記レジスタファイル内の複数の専用レジスタに保持し、該複数の専用レジスタに保持された複数の命令をフェッチしながらリピート命令を実行する、請求項 1 0 記載のデータ処理装置。

【請求項 1 3】 前記レジスタファイルは、プロセッサ状態ワードを含み、

前記メモリ演算部は、リピート命令を実行する場合に、最初のループにおいて前記プロセッサ状態ワード内のフラグをセットし、前記命令メモリからフェッチした前記リピート命令の直後の複数の命令を前記複数の専用レジスタに保持し、

2 回目のループにおいて前記プロセッサ状態ワード内の前記フラグをリセットし、前記複数の専用レジスタに保持された複数の命令をフェッチしながらリピート命令を実行する、請求項 1 2 記載のデータ処理装置。

【請求項 1 4】 前記命令メモリは、複数の命令メモリバンクを含み、

前記メモリ演算部は、前記複数の命令メモリバンクから命令をフェッチする場合に、命令メモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとを発生させてパイプライン処理を行なう、請求項 1 0 ～ 1 3 のいずれかに記載のデータ処理装置。

【請求項 1 5】 前記命令メモリはさらに、連続したアドレスにアクセスした場合に、前記複数の命令メモリバンクの異なる命令メモリバンクにアクセスされるように、下位アドレスを含んだアドレスをデコードして前記複数の命令メモリバンクのチップセレクト信号を生成する第 1 のバンク選択回路を含む、請求項 1 4 記載のデータ処理装置。

【請求項 1 6】 前記データメモリは、複数のデータメモリバンクを含み、前記メモリ演算部は、前記複数のデータメモリバンクにアクセスする場合に、データメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとを発生させてパイプライン処理を行なう、請求項 1 0 ～ 1 5 のいずれかに記載のデータ処理装置。

【請求項 1 7】 前記データメモリはさらに、前記複数のデータメモリバンクを 2 つの異なる領域に分割するために、上位アドレスを含んだアドレスをデコードして前記複数のデータメモリバンクのチップセレクト信号を生成する第 2 のバンク選択回路を含む、請求項 1 6 記載のデータ処理装置。

【請求項 1 8】 前記第 2 のバンク選択回路は、前記 2 つの異なる領域の中の連続したアドレスにアクセスした場合に、前記複数のデータメモリバンクの異なるデータメモリバンクにアクセスされるように、下位アドレスを含んだアドレスをデコードして前記複数のデータメモリバンクのチップセレクト信号を生成する、請求項 1 7 記載のデータ処理装置。

【請求項 1 9】 前記メモリ演算部は、タスクスイッチの切替え時に、前記専用レジスタを含んだ複数のレジスタを退避してタスクスイッチを切替える、請求項 1 0 ～ 1 8 のいずれかに記載のデータ処理装置。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、内部メモリに記憶された命令およびデータをアクセスしながら処理を行なうデータ処理装置に関し、特に、低速で低消費電力のメモリを用いて高速に処理することが可能なデータ処理装置に関する。

【0 0 0 2】

【従来の技術】

近年、CPU (Central Processing Unit) 等のデータ処理装置が広く使用されており、データ処理装置の処理速度の向上に対する要望が高まる一方である。従来のデータ処理装置、たとえばCPUにおいては、命令フェッチ機構、命令デコード機構および命令実行機構などがパイプライン化されている。そして、動作クロックの周波数を高くするとともにメモリのアクセスサイクルを短くし、1パイプラインステージタイム（動作クロックの1クロック）以内にメモリアクセスを行なうことによって、データ処理装置の処理性能を高めている。

【0003】

しかし、アクセス時間の短い大容量メモリを実現するのは困難である。したがって、キャッシュメモリに代表されるように高速で小容量のメモリと、低速で大容量の主メモリとを階層的に構築し、見かけ上高速で大容量のメモリがあるかのように動作させてこの問題を解決していた。この階層化したメモリを用いた従来のデータ処理装置として、たとえば、J.L.Hennessy and D.A.Patterson, "Memory-Hierarchy Design," Computer Architecture a Quantitative Approach, pp.372-483, Morgan kaufmann Publishers, Inc. San Francisco, 1996. に詳しく紹介されている。

【0004】

【発明が解決しようとする課題】

しかし、メモリを階層化することで擬似的に大容量の高速メモリを実現する場合、データ処理装置が小容量の高速メモリに入りきらない命令やデータをアクセスするとき（キャッシュミス）、メモリアクセスサイクルにウェイトサイクルを挿入する必要があり、データ処理装置の性能が低下するという問題点があった。また、メモリを高速動作させるためには、メモリに用いられるトランジスタの駆動能力を大きくする必要があるため、データ処理装置内に大容量の高速メモリを搭載するとメモリの消費電力が増大するという問題点があった。

【0005】

本発明は、上記問題点を解決するためになされたものであり、第1の目的は、低速で低消費電力のメモリを用いつつ高スループットで処理することができ、処

理性能の向上を図ることが可能なデータ処理装置を提供することである。

【0006】

第2の目的は、複数のメモリバンクに分割された命令メモリから命令をフェッチする構成において、リピート命令を実行した場合であっても同じメモリバンクに連続したアクセスが発生することを防止し、処理性能の向上を図ることが可能なデータ処理装置を提供することである。

【0007】

第3の目的は、複数のメモリバンクに分割されたデータメモリにアクセスする構成において、変数データと係数データとを交互に読出す場合などにおいて同じメモリバンクに連続したアクセスが発生することを防止し、処理性能の向上を図ることが可能なデータ処理装置を提供することである。

【0008】

【課題を解決するための手段】

請求項1に記載のデータ処理装置は、命令が格納される命令メモリと、データが格納されるデータメモリと、フェッチされた命令をデコードする命令デコーダと、命令メモリ、データメモリおよび命令デコーダに接続され、命令メモリに格納される命令をフェッチし、命令デコーダのデコード結果に基づいてデータメモリにアクセスするメモリ演算部と、命令デコーダのデコード結果に基づいて整数演算を行なう整数演算部とを含むデータ処理装置であって、命令メモリは、複数の命令メモリバンクを含み、メモリ演算部は、複数の命令メモリバンクから命令をフェッチする場合に、命令メモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとを発生させてパイプライン処理を行なう。

【0009】

メモリ演算部は、命令メモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとを発生させるので、選択した命令メモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となる。また、命令メモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとが並列に行なわれるため、

命令メモリアクセスのスループットを向上させることが可能となる。

【0010】

請求項2に記載のデータ処理装置は、請求項1記載のデータ処理装置であって、命令メモリはさらに、連続したアドレスにアクセスした場合に、複数の命令メモリバンクの異なる命令メモリバンクにアクセスされるように、下位アドレスを含んだアドレスをデコードして複数の命令メモリバンクのチップセレクト信号を生成する第1のバンク選択回路を含む。

【0011】

したがって、同じ命令メモリバンクに連続してアクセスが行なわれなくなり、パイプラインの乱れを防止することが可能となる。

【0012】

請求項3に記載のデータ処理装置は、請求項1または2記載のデータ処理装置であって、命令メモリはさらに、高速命令メモリを含み、メモリ演算部は、高速命令メモリから命令をフェッチする場合に、命令メモリバンクの選択に対応したパイプラインステージを発生させずに、命令の読出しに対応したパイプラインステージを発生させてパイプライン処理を行なう。

【0013】

メモリ演算部は、高速命令メモリから命令をフェッチする場合に、命令メモリバンクの選択に対応したパイプラインステージを発生させないので、高速命令メモリアクセスのレイテンシを向上させることが可能となる。

【0014】

請求項4に記載のデータ処理装置は、請求項1～3のいずれかに記載のデータ処理装置であって、データメモリは、複数のデータメモリバンクを含み、メモリ演算部は、複数のデータメモリバンクにアクセスする場合に、データメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとを発生させてパイプライン処理を行なう。

【0015】

メモリ演算部は、データメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとを発生させるので、選

択したデータメモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となる。また、データメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとが並列に行なわれるため、データメモリアクセスのスループットを向上させることが可能となる。

【 0 0 1 6 】

請求項 5 に記載のデータ処理装置は、請求項 4 記載のデータ処理装置であって、データメモリはさらに、複数のデータメモリバンクを 2 つの異なる領域に分割するために、上位アドレスを含んだアドレスをデコードして複数のデータメモリバンクのチップセレクト信号を生成する第 2 のバンク選択回路を含む。

【 0 0 1 7 】

したがって、変数データと係数データとを交互に読出す場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、パイプラインの乱れを防止することが可能となる。

【 0 0 1 8 】

請求項 6 に記載のデータ処理装置は、請求項 5 記載のデータ処理装置であって、第 2 のバンク選択回路は、2 つの異なる領域の中の連続したアドレスにアクセスした場合に、複数のデータメモリバンクの異なるデータメモリバンクにアクセスされるように、下位アドレスを含んだアドレスをデコードして複数のデータメモリバンクのチップセレクト信号を生成する。

【 0 0 1 9 】

したがって、ポストインクリメントを用いてデータメモリにアクセスする場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、パイプラインの乱れを防止することが可能となる。

【 0 0 2 0 】

請求項 7 に記載のデータ処理装置は、請求項 4 ～ 6 のいずれかに記載のデータ処理装置であって、データメモリはさらに、高速データメモリを含み、メモリ演算部は、高速データメモリにアクセスする場合に、データメモリバンクの選択に対応したパイプラインステージを発生させずに、データのアクセスに対応したパ

イザラインステージを発生させてパイプライン処理を行なう。

【0021】

メモリ演算部は、高速データメモリにアクセスする場合に、データメモリバンクの選択に対応したパイプラインステージを発生させないので、高速データメモリアクセスのレイテンシを向上させることが可能となる。

【0022】

請求項8に記載のデータ処理装置は、請求項1～7のいずれかに記載のデータ処理装置であって、メモリ演算部は、命令バスを介して命令メモリからの命令のフェッチを行ない、命令バスと異なるデータバスを介してデータメモリにアクセスする。

【0023】

したがって、命令メモリからの命令のフェッチと、データメモリへのアクセスとのバスの競合が発生しなくなり、パイプラインの乱れを防止することが可能となる。

【0024】

請求項9に記載のデータ処理装置は、請求項1～8のいずれかに記載のデータ処理装置であって、メモリ演算部は、データ入力バスを介してデータメモリからデータを読出し、データ入力バスと異なるデータ出力バスを介してデータメモリにデータを書込む。

【0025】

したがって、データの読出しと、データの書込みとのバスの競合が発生しなくなり、パイプラインの乱れを防止することが可能となる。

【0026】

請求項10に記載のデータ処理装置は、命令が格納される命令メモリと、データが格納されるデータメモリと、フェッチされた命令をデコードする命令デコーダと、複数のレジスタを有するレジスタファイルと、命令メモリ、データメモリおよび命令デコーダに接続され、命令メモリに格納される命令をフェッチし、命令デコーダのデコード結果に基づいてデータメモリにアクセスするメモリ演算部と、命令デコーダのデコード結果に基づいて整数演算を行なう整数演算部とを含む。

むデータ処理装置であって、メモリ演算部は、リピート命令を実行する場合に、リピート命令の直後の命令をレジスタファイル内の専用レジスタに保持し、専用レジスタに保持された命令をフェッチしながらリピート命令を実行する。

【0027】

メモリ演算部は、専用レジスタに保持された命令をフェッチしながらリピート命令を実行するので、複数のメモリバンクに分割された命令メモリから命令をフェッチする場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、処理性能の向上を図ることが可能となる。

【0028】

請求項11に記載のデータ処理装置は、請求項10記載のデータ処理装置であって、レジスタファイルは、プロセッサ状態ワードを含み、メモリ演算部は、リピート命令を実行する場合に、最初のループにおいてプロセッサ状態ワード内のフラグをセットし、命令メモリからフェッチしたリピート命令の直後の命令を専用レジスタに保持し、2回目のループにおいてプロセッサ状態ワード内のフラグをリセットし、専用レジスタに保持された命令をフェッチしながらリピート命令を実行する。

【0029】

したがって、プロセッサ状態ワードのフラグを参照することにより、リピート命令の実行状況を制御することが可能となる。

【0030】

請求項12に記載のデータ処理装置は、請求項10記載のデータ処理装置であって、メモリ演算部は、リピート命令を実行する場合に、リピート命令の直後の複数の命令をレジスタファイル内の複数の専用レジスタに保持し、複数の専用レジスタに保持された複数の命令をフェッチしながらリピート命令を実行する。

【0031】

メモリ演算部は、複数の専用レジスタに保持された複数の命令をフェッチしながらリピート命令を実行するので、複数のメモリバンクに分割された命令メモリから命令をフェッチする場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、処理性能の向上を図ることが可能となる。

【 0 0 3 2 】

請求項 1 3 に記載のデータ処理装置は、請求項 1 2 記載のデータ処理装置であって、レジスタファイルは、プロセッサ状態ワードを含み、メモリ演算部は、リピート命令を実行する場合に、最初のループにおいてプロセッサ状態ワード内のフラグをセットし、命令メモリからフェッチしたリピート命令の直後の複数の命令を複数の専用レジスタに保持し、2 回目のループにおいてプロセッサ状態ワード内のフラグをリセットし、複数の専用レジスタに保持された複数の命令をフェッチしながらリピート命令を実行する。

【 0 0 3 3 】

したがって、プロセッサ状態ワードのフラグにより、リピート命令の実行状況を制御することが可能となる。

【 0 0 3 4 】

請求項 1 4 に記載のデータ処理装置は、請求項 1 0 ～ 1 3 のいずれかに記載のデータ処理装置であって、命令メモリは、複数の命令メモリバンクを含み、メモリ演算部は、複数の命令メモリバンクから命令をフェッチする場合に、命令メモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとを発生させてパイプライン処理を行なう。

【 0 0 3 5 】

したがって、選択した命令メモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となる。また、命令メモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとが並列に行なわれるため、命令メモリアクセスのスループットを向上させることが可能となる。

【 0 0 3 6 】

請求項 1 5 に記載のデータ処理装置は、請求項 1 4 記載のデータ処理装置であって、命令メモリはさらに、連続したアドレスにアクセスした場合に、複数の命令メモリバンクの異なる命令メモリバンクにアクセスされるように、下位アドレスを含んだアドレスをデコードして複数の命令メモリバンクのチップセレクト信号を生成する第 1 のバンク選択回路を含む。

【 0 0 3 7 】

したがって、同じ命令メモリバンクに連続してアクセスが行なわれなくなり、パイプラインの乱れを防止することが可能となる。

【 0 0 3 8 】

請求項 1 6 に記載のデータ処理装置は、請求項 1 0 ～ 1 5 のいずれかに記載のデータ処理装置であって、データメモリは、複数のデータメモリバンクを含み、メモリ演算部は、複数のデータメモリバンクにアクセスする場合に、データメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとを発生させてパイプライン処理を行なう。

【 0 0 3 9 】

したがって、選択したデータメモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となる。また、データメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとが並列に行なわれるため、データメモリアクセスのスループットを向上させることが可能となる。

【 0 0 4 0 】

請求項 1 7 に記載のデータ処理装置は、請求項 1 6 に記載のデータ処理装置であって、データメモリはさらに、複数のデータメモリバンクを 2 つの異なる領域に分割するために、上位アドレスを含んだアドレスをデコードして複数のデータメモリバンクのチップセレクト信号を生成する第 2 のバンク選択回路を含む。

【 0 0 4 1 】

したがって、変数データと係数データとを交互に読出す場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、パイプラインの乱れを防止することが可能となる。

【 0 0 4 2 】

請求項 1 8 に記載のデータ処理装置は、請求項 1 7 に記載のデータ処理装置であって、第 2 のバンク選択回路は、2 つの異なる領域の中の連続したアドレスにアクセスした場合に、複数のデータメモリバンクの異なるデータメモリバンクにアクセスされるように、下位アドレスを含んだアドレスをデコードして複数のデー

データメモリバンクのチップセレクト信号を生成する。

【0043】

したがって、ポストインクリメントを用いてデータメモリにアクセスする場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、パイプラインの乱れを防止することが可能となる。

【0044】

請求項19に記載のデータ処理装置は、請求項10～18のいずれかに記載のデータ処理装置であって、メモリ演算部は、タスクスイッチの切替え時に、専用レジスタを含んだ複数のレジスタを退避してタスクスイッチを切替える。

【0045】

したがって、元のタスクに復帰する際に、容易にタスクスイッチを切替えることが可能となる。

【0046】

【発明の実施の形態】

図1は、本発明の実施の形態におけるプロセッサを用いたデータ処理装置の概略構成を示すブロック図である。このデータ処理装置は、プロセッサ10と、バス制御回路20と、ROM (Read Only Memory) 21と、SDRAM (Synchronous Dynamic Random Access Memory) 22とを含む。また、プロセッサ10、バス制御回路20、ROM21およびSDRAM22は、アドレスバス31、データバス32および制御バス33によって接続されている。

【0047】

プロセッサ10は、アドレスバス31を介してバス制御回路20、ROM21およびSDRAM22へアドレスを出力する。プロセッサ10がSDRAM22にデータを書込む場合には、データバス32を介してSDRAM22へデータを出力する。プロセッサ10がROM21またはSDRAM22からデータを読み出す場合には、ROM21またはSDRAM22から出力されたデータをデータバス32を介して入力する。また、バス制御回路20は、プロセッサ10から出力される制御信号を受けて、ROM21やSDRAM22を制御するための信号を生成して出力する。

【0048】

図2は、本実施の形態におけるプロセッサ10の概略構成を示すブロック図である。このプロセッサ10は、VLIW.(Very Long Instruction Word)方式のCPUコア（以下、単にコアと呼ぶ。）100と、高速命令メモリ101と、高速データメモリ102と、低電力命令メモリ103と、低電力データメモリ104と、アドレスバス31、データバス32および制御バス33を介して外部のバス制御回路20、ROM21およびSDRAM22に接続されるシステムバスI/F（以下、バスインタフェース部と呼ぶ。）105とを含む。

【0049】

コア100、高速命令メモリ101、高速データメモリ102、低電力命令メモリ103および低電力データメモリ104は、データアドレスバス106、データ出力バス107およびデータ入力バス108に接続されている。また、データと命令とを並列にアクセスできるように、コア100、高速命令メモリ101、低電力命令メモリ103およびバスインタフェース部105は、命令アドレスバス109および命令バス110にも接続されている。

【0050】

また、コア100は、2ウェイのVLIW型命令体系を有しており、2つのサブ命令を含んだVLIW命令を実行する。コア100は、命令バス110を介して入力されたVLIW命令をデコードする命令デコーダ113と、レジスタファイル120と、アドレス演算命令を実行するメモリ演算部130と、整数演算命令を実行する整数演算部140とを含む。

【0051】

命令デコーダ113は、VLIW命令に含まれるそれぞれのサブ命令をデコードするサブ命令デコーダ111および112を含む。メモリ演算部130は、メモリアドレス演算器131と、PC (Program Counter) 演算器132、シフタ133と、ALU (Arithmetic and Logic Unit) 134とを含む。また、整数演算部140は、シフタ141と、ALU142と、乗算器143と、64ビットのアキュムレータ144とを含む。

【0052】

メモリ演算部 1 3 0 および整数演算部 1 4 0 は、2 つのサブ命令デコーダ 1 1 1 および 1 1 2 のデコード結果に従ってそれぞれのサブ命令を実行するが、2 つのサブ命令を並列に実行する場合と、2 つのサブ命令を順番に実行する場合とがある。レジスタファイル 1 2 0 は、ソフトウェアによって読み書き可能な 6 4 本の汎用レジスタを含む。

【 0 0 5 3 】

図 3 は、コア 1 0 0 が有するレジスタを説明するための図である。汎用レジスタ R 0 ~ R 6 3 は、上述したレジスタファイル 1 2 0 内に存在する。レジスタ R 0 は、常に“0”を保持するレジスタであり、他のレジスタのクリア等に使用される。レジスタ R 6 2 は、サブルーチンの戻り先アドレスを保持するためのリンクポインタである。レジスタ R 6 3 は、非割込み処理中におけるスタックポインタおよび割込み処理中におけるスタックポインタであり、後述する P S W (Processor Status Word) 内のモードビット S M によって切替えられる。

【 0 0 5 4 】

アキュムレータ A 0 および A 1 は、乗算結果または積和演算結果を保持するためのレジスタであり、汎用レジスタの 2 倍のビット長である 6 4 ビットのレジスタである。

【 0 0 5 5 】

レジスタ R P T 0 _ C、R P T 0 _ S、R P T 0 _ E および R P T 0 _ I は、リピート命令 R E P E A T 0 に従ってハードウェアループ制御が行なわれるときに使用されるレジスタである。R P T 0 _ C は、ループカウンタ値を保持する。R P T 0 _ S および R P T 0 _ E は、それぞれループの先頭命令のアドレスおよび最後尾命令のアドレスを保持する。R P T 0 _ I は、ループの先頭命令の命令コードを保持する。

【 0 0 5 6 】

レジスタ R P T 1 _ C、R P T 1 _ S、R P T 1 _ E および R P T 1 _ I (n) は、リピート命令 R E P E A T 1 に従ってハードウェアループ制御が行なわれるときに使用されるレジスタである。R P T 1 _ C は、ループカウンタ値を保持する。R P T 1 _ S および R P T 1 _ E は、それぞれループの先頭命令のアドレ

スおよび最後尾命令のアドレスを保持する。RPT1__I (n) は、ループの先頭命令から順に 6 個の命令コードを保持する。

【0057】

レジスタPSWは、プロセッサ状態ワードであり、コア100を制御するためのフラグ等を保持するレジスタである。レジスタPCは、コア100が現在実行しているプログラムのアドレスを保持するレジスタである。レジスタBPSWおよびBPCは、それぞれバックアップ用のPSWおよびPCであり、割り込み等の事象が発生した場合にPSWおよびPCの値が自動的にコピーされる。

【0058】

レジスタMOD0__S、MOD0__E、MOD1__SおよびMOD1__Eは、ループバッファなどに使用するモジュロアドレッシングの制御を行なうためのレジスタである。レジスタMOD0__SおよびMOD0__Eがペアとなって、第1のループバッファの先頭アドレスおよび最後尾アドレスを保持する。また、レジスタMOD1__SおよびMOD1__Eがペアとなって、第2のループバッファの先頭アドレスおよび最後部アドレスを保持する。

【0059】

レジスタIBAは、デバッガがブレークポイントアドレスを指定する際にその値を保持するレジスタである。

【0060】

図4は、PSWの詳細を説明するための図である。SMは、割り込み処理中か非割り込み処理中かを示すビットであり、上述したレジスタR63 (SPU) とR63 (SPI) とを切替えるためのモードビットである。DBは、デバッグ中か否かを示すビットであり、“1”のときに上述したレジスタIBAが有効となる。IEは、割り込み許可中か割り込み禁止中かを示すビットであり、割り込み許可中のときに外部から割り込み要求があるとVLIW命令の切れ目で割り込みを受付ける。

【0061】

RP0およびRP1は、それぞれリピート命令REPEAT0またはREPEAT1の実行によってハードウェアループ制御がイネーブルになると、“1”になるビットである。また、FS0およびFS1は、それぞれリピート命令REP

EAT0またはREPEAT1の実行において、1回目のハードウェアループのときにのみ“1”となるビットである。

【0062】

MD0およびMD1は、それぞれMOD0__SおよびMOD0__Eと、MOD1__SおよびMOD1__Eとによるモジュロアドレッシングをイネーブルにするか、ディスエイブルにするかを決定するビットである。F0～F7は、命令の実行条件を制御するためのビットである。

【0063】

図5は、低電力命令メモリ103の概略構成を示すブロック図である。この低電力命令メモリ103は、8個のメモリバンク40～47と、メモリバンク40～47のいずれかを選択するバンク選択回路48とを含む。バンク選択回路48は、データアドレスバス106、データ出力バス107、命令アドレスバス109および命令バス110に接続される。また、バンク選択回路48は、コア100から出力されるBSI（命令を示すバスステータス）信号およびBSD（データを示すバスステータス）信号を受けて、メモリバンク40～47のCS（チップセレクト）信号を生成する。

【0064】

低電力命令メモリ103は、スループットが1クロックサイクル、レイテンシが2クロックサイクルのメモリである。この理由については後述する。また、コア100からのアドレス転送と読出した命令コードの転送とにそれぞれ1/2クロックサイクルが使用されるので、コア100による命令コードのフェッチにおけるレイテンシが3クロックサイクルとなる。なお、コア100からのアドレス転送および読出した命令コードの転送がメモリアクセスとパイプライン化されているため、スループットは1クロックサイクルのままである。

【0065】

異なるメモリバンクに対して連続してアクセスが行なわれた場合には、スループットが1クロックサイクルとなってコア100が高速に命令をフェッチすることが可能となる。また、同じメモリバンクに対して連続してアクセスが行なわれた場合には、スループットが2クロックサイクルとなって1クロックサイクルの

無駄が発生する。

【0066】

図6は、バンク選択回路48の詳細を説明するためのブロック図である。このバンク選択回路48は、データアドレス信号106が入力されるラッチ56と、命令アドレス信号109が入力されるラッチ57と、ラッチ56および57からの出力を選択するマルチプレクサ55と、アドレス入力レジスタ51と、データ入力レジスタ52と、命令出力レジスタ53と、メモリバンク40～47のCS信号を生成するCS信号生成回路54とを含む。

【0067】

マルチプレクサ55は、ラッチ56によって保持されたデータアドレス信号と、ラッチ57によって保持された命令アドレス信号とを切替えて出力する。アドレス入力レジスタ51は、マルチプレクサ55から出力されたアドレスを保持する。データ入力レジスタ52は、データ出力バス107の内容を保持する。命令出力レジスタ53は、メモリバンク40～47から出力された命令コードを保持して命令バス110へ出力する。

【0068】

アドレス入力レジスタ51、データ入力レジスタ52および命令出力レジスタ53は、それぞれダブルバッファになっている。同じメモリバンクに対して連続してアクセスが行なわれた場合には、スループットが2クロックサイクルとなって1クロックサイクルの無駄が発生する。そのため、2クロックサイクルの間値を保持しつつ、次の命令のアドレス等を保持するためにダブルバッファの構成を採用している。

【0069】

CS信号生成回路54は、BSI信号、BSD信号およびアドレス入力レジスタ51に保持されるアドレス(A0～A16、A27、A28)に基づいて、メモリバンク40～47のCS信号を生成する。上述したように、同じメモリバンクに対して連続してアクセスが行なわれた場合には、スループットが2クロックサイクルとなるため1ウェイト挿入される。WaitI信号およびWaitD信号は、それぞれ命令またはデータのアクセスにおいてウェイトを挿入する際にア

クタイプとなる。

【0070】

図7は、メモリバンクのCS信号の生成を説明するための図である。図7に示すように、32ビットアドレスのうちA0～A15をデコードしてメモリバンク40～47へのアクセスであることを判別する。また、メモリバンク40～47のいずれのメモリバンクへのアクセスであるかを、A16、A27およびA28の3ビットをデコードすることによって判別する。低電力命令メモリ103は8バイト単位でアクセスされるため、A29～A31の値は任意である。なお、図7において、“H'”は16進数であることを示している。

【0071】

図8は、メモリバンク40～47にアクセスするときのパイプライン処理を説明するためのタイミングチャートである。図8は、スループットが1クロックサイクルとなり、レイテンシが2クロックサイクルとなることを説明するためのものである。パイプライン処理の詳細な説明は後述する。

【0072】

図8において、IF0～IF2、D、R、M0～M2およびWは、パイプラインのステージを示している。命令フェッチステージIF0において、最初の1/2クロックで命令アドレスバス109を介して命令アドレスが転送される。次の1/2クロックでメモリバンク40～47の選択が行なわれる。

【0073】

命令フェッチステージIF1において、選択されたメモリバンクのプリチャージが最初の1/2クロックで行なわれる。このタイミングでCS信号生成回路54からメモリバンクのCS信号が出力されて、選択されたメモリバンク内のビット線が活性化される。そして、次の1/2クロックと命令フェッチステージIF2の最初の1/2クロックとで命令のフェッチが行なわれる。なお、命令フェッチステージIF1において、次の命令をフェッチするために次の命令のアドレスバス転送が行なわれ、パイプライン的に処理が行なわれる。

【0074】

命令デコードステージDにおいて、フェッチされた命令がデコードされる。フ

フェッチされた命令がロード命令である場合、リードレジスタステージRにおいてレジスタからデータアドレスが読出される。次に、データメモリアクセスステージM0において、データアドレスバス転送およびバンクの選択が行なわれる。次に、データメモリアクセスステージM1およびM2において、データの読出しおよびデータバス転送が行なわれる。そして、ライトバックステージWにおいて、読出したデータがレジスタに書込まれる。なお、低電力データメモリ104に対するアクセスのタイミングは、低電力命令メモリ103に対するアクセスのタイミングと同じである。

【0075】

通常、コア100が命令をフェッチする場合には、連続したアドレスにアクセスするため、アドレスのA27およびA28が「00」、「01」、「10」、「11」とサイクリックに変化する。したがって、必ず異なるメモリバンクにアクセスすることになり、同じメモリバンクに続けてアクセスすることはない。

【0076】

図9は、低電力データメモリ104の概略構成を示すブロック図である。この低電力データメモリ104は、8個のメモリバンク60～67と、メモリバンク60～67のいずれかを選択するバンク選択回路68とを含む。バンク選択回路68は、データアドレスバス106、データ出力バス107およびデータ入力バス108に接続される。また、バンク選択回路68は、コア100から出力されるBSD信号を受けて、メモリバンク60～67のCS信号を生成する。

【0077】

低電力データメモリ104は、スループットが1クロックサイクル、レイテンシが2クロックサイクルのメモリである。また、コア100からのアドレス転送と読出したデータの転送とにそれぞれ1/2クロックサイクルが使用されるので、コア100によるメモリアクセスにおけるレイテンシが3クロックサイクルとなる。なお、コア100からのアドレス転送および読出したデータの転送がメモリアクセスとパイプライン化されているため、スループットは1クロックサイクルのままである。

【0078】

異なるメモリバンクに対して連続してアクセスが行なわれた場合には、スループットが1クロックサイクルとなってコア100が高速にメモリアクセスを行なうことが可能となる。また、同じメモリバンクに対して連続してアクセスが行なわれた場合には、スループットが2クロックサイクルとなって1クロックサイクルの無駄が発生する。

【0079】

図10は、バンク選択回路68の詳細を説明するためのブロック図である。このバンク選択回路68は、アドレス入力レジスタ71と、データ入力レジスタ72と、データ出力レジスタ73と、メモリバンク60～67のCS信号を生成するCS信号生成回路74とを含む。

【0080】

アドレス入力レジスタ71、データ入力レジスタ72およびデータ出力レジスタ73は、それぞれダブルバッファになっている。同じメモリバンクに対して連続してアクセスが行なわれた場合には、スループットが2クロックサイクルとなって1クロックサイクルの無駄が発生する。そのため、2クロックサイクルの間値を保持しつつ、次のデータのアドレス等を保持するためにダブルバッファの構成を採用している。

【0081】

CS信号生成回路74は、BSD信号およびアドレス入力レジスタ71に保持されるアドレス(A0～A16、A27、A28)に基づいて、メモリバンク60～67のCS信号を生成する。上述したように、同じメモリバンクに対して連続してアクセスが行なわれた場合には、スループットが2クロックサイクルとなるため1ウェイト挿入される。WaitD信号は、メモリアクセスにおいてウェイトを挿入する際にアクティブとなる。

【0082】

図11は、メモリバンクのCS信号の生成を説明するための図である。図11に示すように、32ビットアドレスのうちA0～A15をデコードしてメモリバンク60～67へのアクセスであることを判別する。また、メモリバンク60～67のいずれのメモリバンクへのアクセスであるかを、A16、A27およびA

28の3ビットをデコードすることによって判別する。A16は、メモリバンク60～63へのアクセスであるか、メモリバンク64～67へのアクセスであるかを示している。低電力データメモリ104は8バイト単位でアクセスされるため、A29～A31の値は任意である。

【0083】

デジタル信号処理においては、連続したアドレスのデータが順次アクセスされる場合が多い。8バイト単位でデータがアクセスされる場合には、アドレスのA27およびA28が「00」、「01」、「10」、「11」とサイクリックに変化する。したがって、必ず異なるメモリバンクにアクセスすることになり、同じメモリバンクに続けてアクセスすることはない。

【0084】

また、1、2、4バイト単位でデータがアクセスされる場合には、最初のメモリアクセスで一旦8バイトのデータがデータ出力レジスタ73に格納される。そして、2回目以降のメモリアクセスにおいては、メモリバンクへのアクセスは行なわれずに、データ出力レジスタ73に格納されたデータが順次データ入力バス108に出力される。したがって、スループットが1に保たれる。

【0085】

デジタル信号処理においては、低電力データメモリ104から同数の変数と係数とが読出される場合が多い。したがって、アドレスのA16の値によって変数が格納される領域（バンクメモリ60～63）と係数が格納される領域（バンクメモリ64～67）とを分け、変数と係数とを交互に読出す場合でも同じバンクメモリに対するアクセスが発生しないようにしている。

【0086】

図12は、コア100が実行する命令のフォーマットを説明するための図である。コア100が実行する命令は、2ウェイのVLIW型命令であり、サブ命令を格納するLコンテナ205およびRコンテナ206と、各サブ命令の実行条件を指定するCCフィールド203および204と、各サブ命令の実行順序または長いサブ命令を定義するFMフィールド201aおよび201bとを含む。

【0087】

CCフィールド203および204は、PSW中のフラグF0およびF1に依存した条件を指定する。たとえば、CCフィールド203が“000”のとき、Lコンテナ205に含まれるサブ命令が無条件に実行される。また、CCフィールド204が“101”のとき、Rコンテナ206に含まれるサブ命令が、フラグF0=“1”かつF1=“1”の場合に実行され、フラグF0およびF1がそれ以外の値の場合にはサブ命令が無効化される。

【0088】

FMフィールド201aおよび201bは、Lコンテナ205およびRコンテナ206に含まれるサブ命令の実行順序または長いサブ命令を定義する。FMフィールドが“00”のとき、Lコンテナ205およびRコンテナ206に含まれる2つのサブ命令が並列に実行される。FMフィールドが“01”のとき、Lコンテナ205に含まれるサブ命令がまず実行され、次にRコンテナ206に含まれるサブ命令が実行される。FMフィールドが“10”のとき、Rコンテナ206に含まれるサブ命令がまず実行され、次にLコンテナ205に含まれるサブ命令が実行される。FMフィールドが“11”のとき、Lコンテナ205およびRコンテナ206に分割して保持された1つの長いサブ命令が実行される。

【0089】

図13(a)～図13(h)は、Lコンテナ205およびRコンテナ206に保持されるサブ命令のフォーマットを示す図である。短いサブ命令は28ビットの長さを有し、7種類のフォーマットに分類される。図13(a)～図13(g)に示すように、短いサブ命令のビット位置0～9で演算の種類が指定され、ビット位置10～27で最大3つのオペランドが指定される。長いサブ命令は54ビットの長さを有し、図13(h)に示すように、長いサブ命令のビット位置0～9で演算の種類が指定され、ビット位置10～53で32ビット長の即値データを含む最大3つのオペランドが指定される。なお、長いサブ命令の32ビットの即値は、図12に示すVLIW命令におけるビット位置26～31、36～43および46～63に保持される。

【0090】

図13(a)に示すフォーマットは、メモリアクセス演算(LD命令/ST命

命令)を行なうサブ命令のフォーマットである。このサブ命令は、演算内容（オペコード）を指定するフィールド（ビット位置0～7）と、レジスタであるか即値であるかを指定するフィールドX（ビット位置8～9）、レジスタ番号を指定するフィールドR a（ビット位置10～15）およびR b（ビット位置16～21）と、レジスタ番号または6ビット長の即値を指定するフィールドs r c（ビット位置22～27）とを含む。図13（a）に示すように、フィールドXの値が“00”、“01”または“11”である場合は、フィールドs r cがレジスタ番号であることを示しており、フィールドXの値が“10”である場合は、フィールドs r cが即値であることを示している。このサブ命令は、レジスタ間接アドレッシングによるメモリアクセス演算に用いられる。なお、“R b++”および“R b--”は、レジスタ間接アドレッシングの際のアドレッシングモードを示しており、“R b++”はポストインクリメント付きレジスタ間接モードを、“R b--”はポストデクリメント付きレジスタ間接モードを示している。

【0091】

図13（b）に示すフォーマットは、汎用レジスタに保持されたオペランド間の演算（ALU命令）または積和演算（MAC命令）を行なうサブ命令のフォーマットである。このサブ命令は、レジスタ番号または即値を指定するフィールドY（ビット位置8）を含む。図13（b）に示すように、フィールドYの値が“0”である場合は、s r cがレジスタ番号であることを示しており、フィールドYの値が“1”である場合は、s r cが即値であることを示している。

【0092】

図13（c）～図13（g）に示すフォーマットは、ブランチ演算（BRA命令）を行なうサブ命令のフォーマットである。図13（c）～図13（g）に示すように、フィールドF（ビット位置8）によってレジスタであるか、即値であるかが指定され、フィールドZ（ビット位置9）によってゼロフラグを参照した分岐命令であることが指定される。また、ビット位置10～27に示すフィールドによって、レジスタまたは即値による分岐先アドレスが指定される。

【0093】

図13（h）に示すフォーマットは、長いサブ命令のフォーマットを示してい

る。このサブ命令は、演算内容（オペコード）を指定するフィールド（ビット位置0～7）と、レジスタ番号を指定するフィールドR a（ビット位置10～15）およびR b（ビット位置16～21）と、32ビット長の即値を指定するフィールドi m m（ビット位置22～53）とを含む。この長いサブ命令は、メモリアクセス演算、汎用レジスタに保持されたオペランド間の演算、積和演算およびブランチ演算のいずれにも使用される。

【0094】

図14は、本実施の形態におけるコア100のパイプライン処理を説明するための図である。図14（a）～図14（d）は、低電力命令メモリ103からフェッチした命令を実行するときのパイプライン処理を示しており、それぞれALU命令、MAC命令、LD／ST命令およびBRA命令を示している。ステージIF0～IF2は、命令フェッチステージを示している。ステージDは、命令デコードステージを示している。

【0095】

ステージRは、レジスタファイル120からオペランドを読出すステージである。ステージR／Aは、レジスタファイル120からPC値の読出し、または読出したPC値にディスプレースメント値を加算して分岐先アドレスを計算するステージである。ステージE0およびE1は、命令実行のための演算を行なうステージである。ステージM0～M2は、データメモリに対してアクセスを行なうステージである。ステージWは、オペランドを汎用レジスタに書込むステージである。

【0096】

図14（e）～図14（h）は、高速命令メモリ101からフェッチした命令を実行するときのパイプライン処理を示しており、それぞれALU命令、MAC命令、LD／ST命令およびBRA命令を示している。これらのパイプライン処理は、図14（a）～図14（d）に示すパイプライン処理と比較して、命令フェッチステージがIF0およびIF1の2段となっている点、およびデータメモリアクセスステージがM0およびM1の2段となっている点異なる。このデータメモリアクセスステージが2段となっているのは、高速データメモリ102に

アクセスする場合であり、低電力データメモリ104にアクセスする場合にはM0～M2の3段となる。

【0097】

図14(c)および図14(g)に示すように、LD/ST命令の実行において、ステージM0～M2またはM0～M1の複数のパイプラインステージでデータメモリにアクセスする場合、データバスにおけるデータ転送の競合が発生するため、たとえばロード命令の実行直後にストア命令の実行ができないという問題点が従来あった。しかし、本実施の形態におけるコア100は、ロード命令実行時にデータ入力バス108を介してデータを転送し、ストア命令実行時にデータ出力バス107を介してデータを転送するので、ロード命令の実行直後にストア命令を実行する場合であってもパイプラインが乱れることはない。また、それぞれのデータバスはデータの転送方向が一定であるので、回路が簡単になるというメリットがある。

【0098】

上述したように、コア100が高速メモリ101または102にアクセスする場合と、低電力メモリ103または104にアクセスする場合とで、パイプライン処理の段数を可変にしている。低電力命令メモリ103からフェッチした命令を実行する場合のパイプライン処理におけるタイミングについては、図8を用いて説明した。

【0099】

図15は、高速命令メモリ101からフェッチした命令を実行する場合のパイプライン処理を説明するためのタイミングチャートである。命令フェッチステージIF0において、最初の1/2クロックで命令フェッチのためのアドレスバス転送と、プリチャージが同時に行なわれる。このプリチャージは、アドレスの値にかかわらず高速命令メモリ101にアクセスする際に必ず行なわれるため、低電力命令メモリ103にアクセスする場合と比較して、1クロックサイクルだけ早くアクセスが終了する。

【0100】

そして、次の1/2クロックと命令フェッチステージIF1の最初の1/2ク

・ ロックとで命令のフェッチが行なわれる。なお、命令フェッチステージ I F 1 において、次の命令をフェッチするために次の命令のアドレスバス転送が行なわれ、パイプライン的に処理が行なわれる。

【0101】

命令デコードステージ D において、フェッチされた命令がデコードされる。フェッチされた命令がロード命令である場合、リードレジスタステージ R においてレジスタからデータアドレスが読出される。次に、データメモリアクセスステージ M 0 において、データアドレスバス転送と同時に高速データメモリ 102 に対するプリチャージが行なわれる。このプリチャージは、アドレスの値にかかわらず高速データメモリ 102 にアクセスする際に必ず行なわれるため、低電力データメモリ 104 にアクセスする場合と比較して、1 クロックサイクルだけ早くアクセスが終了する。

【0102】

ステージ M 0 の次の 1/2 クロックおよび M 1 において、データの読出しおよびデータバス転送が行なわれる。そして、ライトバックステージ W において、読出したデータがレジスタに書込まれる。

【0103】

図 16 は、ロード／ストア命令、データ転送命令および比較命令の一覧を示す図である。図 17 は、算術演算命令、論理演算命令、シフト演算命令およびビット演算命令の一覧を示す図である。図 18 は、分岐命令、O S (Operating System) 関連命令、D S P (Digital Signal Processor) 関連命令、リピート命令およびデバッグ支援命令の一覧を示す図である。図 16～図 18 において、大文字で各サブ命令のニーモニックが記載され、その横にサブ命令のオペレーションの内容が記載されている。各サブ命令のオペレーションの内容は、図 16～図 18 に詳細に記載されているので、詳細な説明は行なわない。

【0104】

図 19 は、本実施の形態におけるプロセッサ 10 のメモリマップの一例を示す図である。図 19 に示すように、各メモリはアドレス値によって区別されており、コア 100 はアドレス値によってメモリアクセス開始からメモリアクセス終了

までのサイクル数、すなわちパイプラインの段数を決定する。なお、図2に示すように、高速命令メモリ101および低電力命令メモリ103からの命令のフェッチと、高速データメモリ102および低電力データメモリ104に対するアクセスとが異なるバスで行なわれるため、バスの競合が発生することはない。

【0105】

デジタル信号処理においては、FIR (Finite Impulse Response) フィルタ等の処理を行なう際にループが多用される。本実施の形態のコア100においては、ループ処理をハードウェアで行ない、ループのための分岐を実質的にゼロクロックサイクルで処理するためのリピート命令 (REPEAT0, REPEAT1) をサポートしている。以下、このリピート命令について詳細に説明する。

【0106】

図20は、REPEAT1のオペレーションとそれを用いたFIRフィルタ処理のプログラムの一例とを示す図である。REPEAT1のオペレーションに示すように、ループの回数countと、ループの最初の命令から最後の命令までの距離pcaddrとが指定され、RPT1__C、RPT1__S、RPT1__EおよびRPT1__I (0:5) の各レジスタに、それぞれループ回数、ループ開始アドレス (REPEAT1命令の直後の命令アドレス)、ループ終了アドレス、およびループの先頭から6個の命令が入力され、PSWのRP1ビットおよびFS1ビットがセットされる。そして、ループの回数countだけ、ループの最初の命令から最後の命令までが実行される。

【0107】

図20に示すプログラム例においては、ループの回数countに“20”が設定され、ループの最初の命令から最後の命令までの距離pcaddrに“48”が設定される。そして、STARTからENDまでの処理、すなわちLD2W命令とMAC0命令を並列に実行するVLIW命令6個からなる処理が20回繰返される。

【0108】

図21は、図20に示すプログラムを実行したときのパイプライン処理を説明するための図である。高速命令メモリ101からLD2W命令およびMAC0命

- ・ 令がフェッチされ、メモリ演算部130および整数演算部140が、LD2W命令およびMAC0命令を並列にパイプライン処理し、1クロックサイクルに1回ずつ積和演算結果がアキュムレータA0に格納されるところを示している。

【0109】

REPEAT1命令実行直後の1回目のループにおいてはFS1ビットがセットされ、STARTからENDまでのVLIW命令が実行されるとともに、その6個のVLIW命令が6個のレジスタRPT1_I(0:5)に書込まれる。1回目のループの最後の分岐においてFS1ビットがリセットされ、2回目のループ以降最後のループまで、レジスタRPT1_I(0:5)に格納されたVLIW命令がフェッチされて高速命令メモリ101へのアクセスは行なわれない。ループの実行回数に応じて、レジスタRPT1_Cの値が1ずつデクリメントされ、RPT1_Cの値が“0”になるとループが終了してRP1ビットがリセットされる。

【0110】

また、ループ実行中にプロセッサ10が割込み要求を受付けると、このプログラムの実行が中断され、現タスクAが新タスクBにスイッチするが、タスクBを実行する前にOSがBPSWに退避されたタスクA実行中のPSW値と、RTP1_C、RTP1_S、RTP1_EおよびRTP1_I(0:5)の各レジスタを含むタスクAの実行環境をすべてメモリに退避し、タスクAの処理に戻るときにOSがタスクAの実行環境を復帰させる。そのため、REPEAT1命令によって起動されたループ処理がタスクスイッチの切替えによって壊れることはない。

【0111】

図22は、REPEAT0のオペレーションとそれを用いたFIRフィルタ処理のプログラムの一例とを示す図である。REPEAT0のオペレーションに示すように、ループの回数countと、ループの最初の命令から最後の命令までの距離pcaddrとが指定され、RPT0_C、RPT0_S、RPT0_EおよびRPT0_Iの各レジスタに、それぞれループ回数、ループ開始アドレス(REPEAT0命令の直後の命令アドレス)、ループ終了アドレス、およびル

ープの先頭命令が入力され、PSWのRPOビットおよびFSOビットがセットされる。そして、ループの回数countだけ、ループの最初の命令から最後の命令までが実行される。

【0112】

図22に示すプログラム例においては、ループの回数countに“10”が設定され、ループの最初の命令から最後の命令までの距離pcaddrに“64”が設定される。そして、STARTからENDまでの処理、すなわちLD2W命令とMAC0命令とを並列に実行するVLIW命令8個からなる処理が10回繰返される。

【0113】

図23は、図22に示すプログラムを実行したときのパイプライン処理を説明するための図である。低電力命令メモリ103からLD2W命令およびMAC0命令がフェッチされ、メモリ演算部130および整数演算部140が、LD2W命令およびMAC0命令を並列にパイプライン処理し、1クロックサイクルに1回ずつ積和演算結果がアキュムレータA0に格納されるところを示している。

【0114】

REPEAT0命令実行直後の1回目のループにおいてはFSOビットがセットされ、STARTからENDまでのVLIW命令が実行されるとともに、最初のVLIW命令（ラベルSTARTが付されたVLIW命令）がレジスタRPT0__Iに書込まれる。1回目のループの最後の分岐においてFSOビットがリセットされ、2回目のループ以降最後のループまで、レジスタRPT0__Iに格納された最初のVLIW命令および低電力命令メモリ103に格納された最初のVLIW命令以外の命令がフェッチされて実行される。ループの実行回数に応じて、レジスタRPT0__Cの値が1ずつデクリメントされ、RPT0__Cの値が“0”になるとループが終了してRPOビットがリセットされる。

【0115】

図22に示す8個のVLIW命令が低電力命令メモリ103のメモリバンク40～47に保持されており、1回目のループにおいてはメモリバンク40から順番にVLIW命令がアクセスされて、1クロックサイクル毎にVLIW命令がフ

エッチされる。図 2 2 に示すプログラム例の場合には、V L I W 命令が 8 個であるので連続して同じメモリバンクにアクセスすることがなく、スループットが 1 となる。しかし、ループ内の V L I W 命令の数が (バンク数 \times n + 1) の場合には、ループ内の最後の命令と最初の命令とが同じメモリバンクに存在することになり、1 ウェイト挿入されてスループットが 2 となる。

【 0 1 1 6 】

一方、本実施の形態におけるプロセッサ 1 0 においては、ループの最初の命令がレジスタ R P T 0 _ I に保持されているため、連続して同じメモリバンクにアクセスされることがなくなり、図 2 3 に示すようにパイプラインに乱れが発生することはない。

【 0 1 1 7 】

図 2 4 は、図 2 2 に示すプログラムのループ部分が実行されるときデータの流れを示す図である。メモリ演算部 1 3 0 および整数演算部 1 4 0 は、L D 2 W 命令および M A C 0 命令を並列に実行する。メモリ演算部 1 3 0 は、低電力データメモリ 1 0 4 の係数領域 3 0 2 (アドレス H' 8000 8100 ~ H' 8000 8128) からレジスタ R 1 0 ~ R 1 7 へ係数を 2 個ずつロードする L D 2 W 命令と、低電力データメモリ 1 0 4 の変数領域 3 0 1 (アドレス H' 8000 0100 ~ H' 8000 0128) からレジスタ R 2 0 ~ R 2 7 へ変数を 2 個ずつロードする L D 2 W 命令とが交互に実行される。

【 0 1 1 8 】

また、メモリ演算部 1 3 0 は、レジスタ R 3 0 または R 3 1 に保持しているポインタを命令実行毎に 8 ずつポストインクリメントし、連続する係数および変数を次々にレジスタファイル 1 2 0 へロードする。メモリ演算部 1 3 0 は、奇数番目の L D 2 W 命令を実行することによって、メモリバンク 6 0 ~ 6 3 に保持される係数データをレジスタ R 1 0 ~ R 1 7 にロードする。また、メモリ演算部 1 3 0 は、偶数番目の L D 2 W 命令を実行することによって、メモリバンク 6 4 ~ 6 7 に保持される変数データをレジスタ R 2 0 ~ R 2 7 にロードする。

【 0 1 1 9 】

図 1 1 に示すように、低電力データメモリ 1 0 4 のメモリバンク 6 0 ~ 6 7 の

CS信号を生成する際に、A16、A27およびA28をデコードしている。したがって、低電力データメモリ104のメモリバンク60～63が変数領域301となり、メモリバンク64～67が係数領域302となるため、図22に示すLD2W命令を連続して実行した場合であっても、同じメモリバンクにアクセスされることがなくパイプラインに乱れが発生することはない。

【0120】

また、たとえ連続して係数をアクセスする場合や、連続して変数をアクセスする場合であっても、A27およびA28をデコードしてメモリバンクのCS信号を生成しているため、連続する係数や変数が同じメモリバンクに存在することは起こり得ない。したがって、ポストインクリメントアドレッシングモードを使用する限り、同じバンクへ連続してアクセスされることはない。

【0121】

本実施の形態においては、低電力命令メモリ103の同じメモリバンクに連続アクセスが行なわれないプログラム例として、FIRフィルタ処理について説明した。しかし、命令のフェッチにおいて分岐が発生しない限り連続するアドレスにアクセスが行なわれる。したがって、下位アドレスをデコードして2つ以上のメモリバンクに分割すれば、分岐以外の命令実行において同じメモリバンクに連続してアクセスが発生することはない。

【0122】

また、デジタル信号処理において、FIRフィルタ処理以外でも連続したアドレス領域に係数や変数を格納して順次アクセスする場合、下位アドレスをデコードして低電力データメモリ104を2つ以上のメモリバンクに分割しておけば、同じメモリバンクに連続してアクセスが発生することはない。

【0123】

以上説明したように、本実施の形態におけるデータ処理装置によれば、メモリ演算部130が低電力命令メモリ103のメモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとを発生させるので、選択したメモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となった。また、パイプラインステージIF0～IF2が並

列に行なわれるため、低電力命令メモリ103のスループットを向上させることが可能となった。

【0124】

また、バンク選択回路48は、下位アドレスA27およびA28をデコードしてメモリバンク40～47のCS信号を生成するので、メモリ演算部130が連続したアドレス領域から命令をフェッチする場合でも異なるメモリバンクにアクセスが行なわれるため、パイプラインに乱れが発生することを防止できる。また、高速命令メモリ101から命令をフェッチする場合には、バンクセレクトを行わず、プリチャージとアドレス転送とを並列して行なうようにしたので、高速に命令をフェッチすることが可能となった。

【0125】

メモリ演算部130が低電力データメモリ104のメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとを発生させるので、選択したメモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となった。また、パイプラインステージM0～M2が並列に行なわれるため、低電力データメモリ104のスループットを向上させることが可能となった。

【0126】

また、バンク選択回路68は、下位アドレスA27およびA28をデコードしてメモリバンク60～67のCS信号を生成するので、メモリ演算部130が連続したアドレス領域にデータのアクセスを行なう場合でも異なるメモリバンクにアクセスが行なわれるため、パイプラインに乱れが発生することを防止できる。また、高速データメモリ102にデータのアクセスを行なう場合には、バンクセレクトを行わず、アドレスの転送とプリチャージとを並列して行なうようにしたので、高速にデータのアクセスを行なうことが可能となった。

【0127】

また、バンク選択回路68は、上位アドレスA16をデコードしてメモリバンク60～67のCS信号を生成するので、係数と変数とを別々の領域に格納することによって、係数と変数とを交互に読み出す場合でも同じメモリバンクに連続

してアクセスが発生することがなくなり、パイプラインに乱れが発生することを防止できる。

【0128】

メモリ演算部130は、リピート命令を実行する場合に、レジスタRPT0__IまたはRPT1__I(n)にリピート命令の直後の命令を保持するようにしたので、ループの最後の命令からループの最初の命令に分岐する場合であっても、同じメモリバンクに連続してアクセスすることがなくなり、処理性能の低下を防止することが可能となった。

【0129】

また、ループ実行中にタスクスイッチが切替えられる場合であっても、OSがBPSWに退避されたPSW値と、RTP1__C、RTP1__S、RTP1__EおよびRTP1__I(0:5)の各レジスタを含むタスクの実行環境をすべてメモリに退避するようにしたので、REPEAT命令によって起動されたループ処理がタスクスイッチの切替えによって壊れるのを防止することが可能となった。

【0130】

また、最初のループにおいてFS1ビットがセットされ、2回目以降のループにおいてFS1ビットがリセットされるので、リピート命令の実行状況を容易に把握することが可能となった。

【0131】

今回開示された実施の形態は、すべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【0132】

【発明の効果】

請求項1に記載のデータ処理装置によれば、メモリ演算部が命令メモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとを発生させるので、選択した命令メモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となった。また、命令メモリバ

ンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとが並列に行なわれるため、命令メモリのスループットを向上させることが可能となった。

【 0 1 3 3 】

請求項 2 に記載のデータ処理装置によれば、同じ命令メモリバンクに連続してアクセスが行なわれなくなり、パイプラインの乱れを防止することが可能となった。

【 0 1 3 4 】

請求項 3 に記載のデータ処理装置によれば、メモリ演算部が高速命令メモリから命令をフェッチする場合に、高速命令メモリに対して少ないパイプラインステージ数でアクセスを行なうので、高速命令メモリのレイテンシを向上させることが可能となった。

【 0 1 3 5 】

請求項 4 に記載のデータ処理装置によれば、メモリ演算部がデータメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとを発生させるので、選択したデータメモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となった。また、データメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとが並列に行なわれるため、データメモリのスループットを向上させることが可能となった。

【 0 1 3 6 】

請求項 5 に記載のデータ処理装置によれば、変数データと係数データとを交互に読出す場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、パイプラインの乱れを防止することが可能となった。

【 0 1 3 7 】

請求項 6 に記載のデータ処理装置によれば、ポストインクリメントを用いてデータメモリにアクセスする場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、パイプラインの乱れを防止することが可能となった。

【 0 1 3 8 】

請求項 7 に記載のデータ処理装置によれば、メモリ演算部が高速データメモリにアクセスする場合に、高速データメモリに対して少ないパイプラインステージ数でアクセスを行なうので、高速データメモリのレイテンシを向上させることが可能となった。

【 0 1 3 9 】

請求項 8 に記載のデータ処理装置によれば、命令メモリからの命令のフェッチと、データメモリへのアクセスとのバスの競合が発生しなくなり、パイプラインの乱れを防止することが可能となった。

【 0 1 4 0 】

請求項 9 に記載のデータ処理装置によれば、データの読出しと、データの書込みとのバスの競合が発生しなくなり、パイプラインの乱れを防止することが可能となった。

【 0 1 4 1 】

請求項 1 0 に記載のデータ処理装置によれば、メモリ演算部が専用レジスタに保持された命令をフェッチしながらリピート命令を実行するので、複数のメモリバンクに分割された命令メモリから命令をフェッチする場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、処理性能の向上を図ることが可能となった。

【 0 1 4 2 】

請求項 1 1 に記載のデータ処理装置によれば、プロセッサ状態ワードのフラグにより、リピート命令の実行状況を制御してマルチタスクにも対応することが可能となった。

【 0 1 4 3 】

請求項 1 2 に記載のデータ処理装置によれば、メモリ演算部が複数の専用レジスタに保持された複数の命令をフェッチしながらリピート命令を実行するので、複数のメモリバンクに分割された命令メモリから命令をフェッチする場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、処理性能の向上を図ることが可能となった。

【 0 1 4 4 】

請求項 1 3 に記載のデータ処理装置によれば、プロセッサ状態ワードのフラグにより、リピート命令の実行状況を制御してマルチタスクにも対応することが可能となった。

【 0 1 4 5 】

請求項 1 4 に記載のデータ処理装置によれば、選択した命令メモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となった。また、命令メモリバンクの選択に対応したパイプラインステージと、命令の読出しに対応したパイプラインステージとが並列に行なわれるため、命令メモリのスループットを向上させることが可能となった。

【 0 1 4 6 】

請求項 1 5 に記載のデータ処理装置によれば、同じ命令メモリバンクに連続してアクセスが行なわれなくなり、パイプラインの乱れを防止することが可能となった。

【 0 1 4 7 】

請求項 1 6 に記載のデータ処理装置によれば、選択したデータメモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となった。また、データメモリバンクの選択に対応したパイプラインステージと、データのアクセスに対応したパイプラインステージとが並列に行なわれるため、データメモリのスループットを向上させることが可能となった。

【 0 1 4 8 】

請求項 1 7 に記載のデータ処理装置によれば、変数データと係数データとを交互に読出す場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、パイプラインの乱れを防止することが可能となった。

【 0 1 4 9 】

請求項 1 8 に記載のデータ処理装置によれば、ポストインクリメントを用いてデータメモリにアクセスする場合であっても、同じメモリバンクに連続したアクセスが発生しなくなり、パイプラインの乱れを防止することが可能となった。

【 0 1 5 0 】

請求項 1 9 に記載のデータ処理装置によれば、元のタスクに復帰する際に、容

易にタスクスイッチを切替えることが可能となった。

【図面の簡単な説明】

【図 1】 本発明の実施の形態におけるプロセッサ 1 0 を用いたデータ処理装置の概略構成を示すブロック図である。

【図 2】 本発明の実施の形態におけるプロセッサ 1 0 の概略構成を示すブロック図である。

【図 3】 コア 1 0 0 が有するレジスタを説明するための図である。

【図 4】 P S W の詳細を説明するための図である。

【図 5】 低電力命令メモリ 1 0 3 の概略構成を示すブロック図である。

【図 6】 バンク選択回路 4 8 の詳細を説明するためのブロック図である。

【図 7】 メモリバンクの C S 信号の生成を説明するための図である。

【図 8】 メモリバンク 4 0 ～ 4 7 にアクセスするときのパイプライン処理を説明するための図である。

【図 9】 低電力データメモリ 1 0 4 の概略構成を示すブロック図である。

【図 1 0】 バンク選択回路 6 8 の詳細を説明するためのブロック図である。

【図 1 1】 メモリバンクの C S 信号の生成を説明するための図である。

【図 1 2】 コア 1 0 0 が実行する命令のフォーマットを説明するための図である。

【図 1 3】 L コンテナ 2 0 5 および R コンテナ 2 0 6 に保持されるサブ命令のフォーマットを示す図である。

【図 1 4】 本発明の実施の形態におけるコア 1 0 0 のパイプライン処理を説明するための図である。

【図 1 5】 高速命令メモリ 1 0 1 からフェッチした命令を実行する場合のパイプライン処理を説明するための図である。

【図 1 6】 ロード／ストア命令、データ転送命令および比較命令の一覧を示す図である。

【図 1 7】 算術演算命令、論理演算命令、シフト演算命令およびビット演算命令の一覧を示す図である。

【図 1 8】 分岐命令、OS 関連命令、DSP 関連命令、リピート命令およびデバッガ支援命令の一覧を示す図である。

【図 1 9】 本発明の実施の形態におけるプロセッサ 1 0 のメモリマップの一例を示す図である。

【図 2 0】 REPEAT 1 命令のオペレーションとそれを用いた FIR フィルタ処理のプログラムの一例とを示す図である。

【図 2 1】 図 2 0 に示すプログラムを実行したときのパイプライン処理を説明するための図である。

【図 2 2】 REPEAT 0 命令のオペレーションとそれを用いた FIR フィルタ処理のプログラムの一例とを示す図である。

【図 2 3】 図 2 2 に示すプログラムを実行したときのパイプライン処理を説明するための図である。

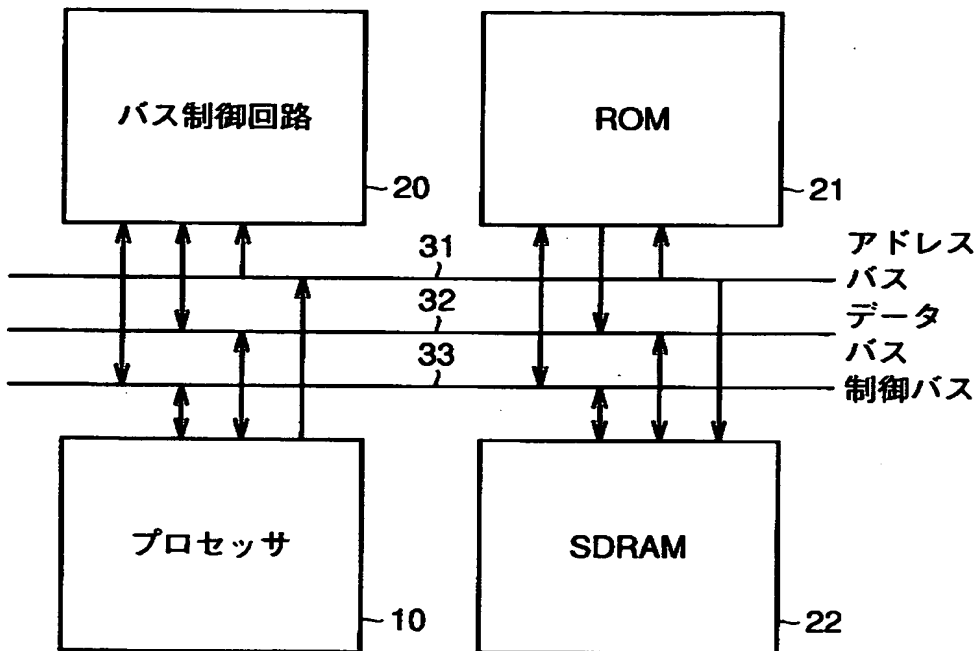
【図 2 4】 図 2 2 に示すプログラムのループ部分が実行されるときデータの流れを示す図である。

【符号の説明】

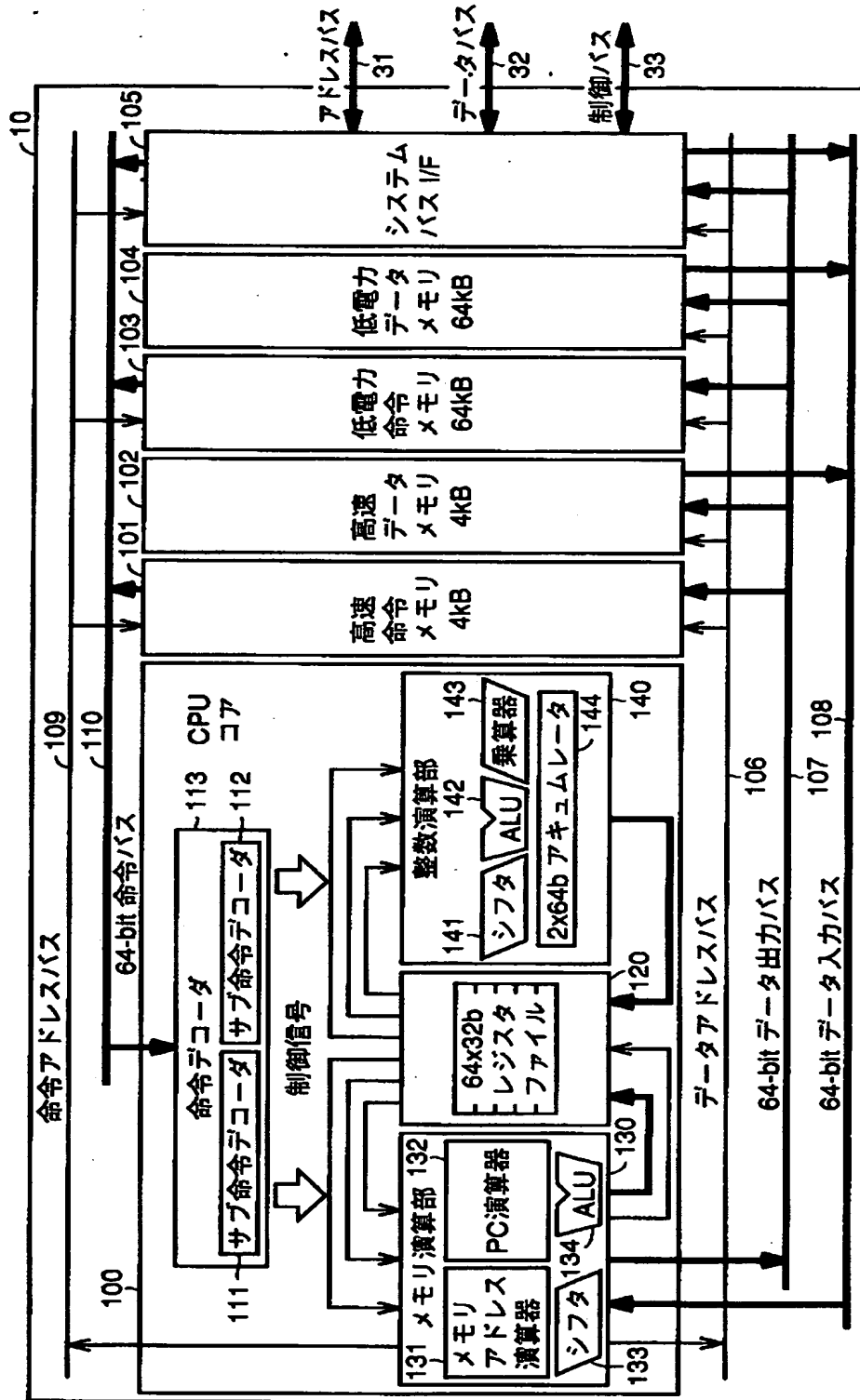
1 0 プロセッサ、2 0 バス制御回路、2 1 ROM、2 2 SDRAM、4 0 ~ 4 7, 6 0 ~ 6 7 メモリバンク、4 8, 6 8 バンク選択回路、5 1, 7 1 アドレス入力レジスタ、5 2, 7 2 データ入力レジスタ、5 3 命令出力レジスタ、5 4, 7 4 CS 信号生成回路、5 5 マルチプレクサ、5 6, 5 7 ラッチ、7 3 データ出力レジスタ、1 0 0 CPU コア、1 0 1 高速命令メモリ、1 0 2 高速データメモリ、1 0 3 低電力命令メモリ、1 0 4 低電力データメモリ、1 1 1, 1 1 2 サブ命令デコーダ、1 1 3 命令デコーダ、1 2 0 レジスタファイル、1 3 0 メモリ演算部、1 3 1 メモリアドレス演算器、1 3 2 PC 演算器、1 3 3, 1 4 1 シフタ、1 3 4, 1 4 2 ALU、1 4 0 整数演算部、1 4 3 乗算器、1 4 4 アキュムレータ。

【書類名】 図面

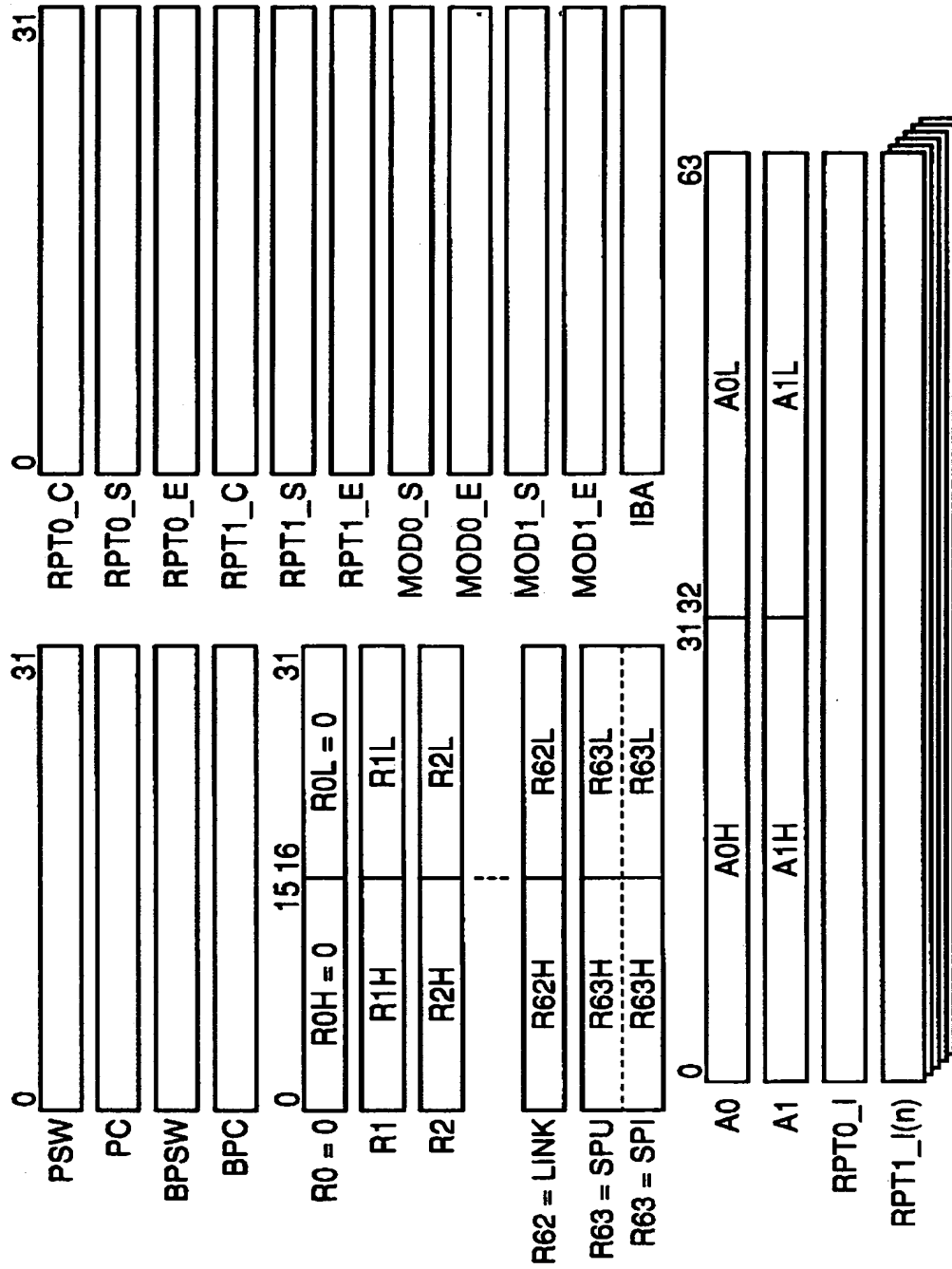
【図1】



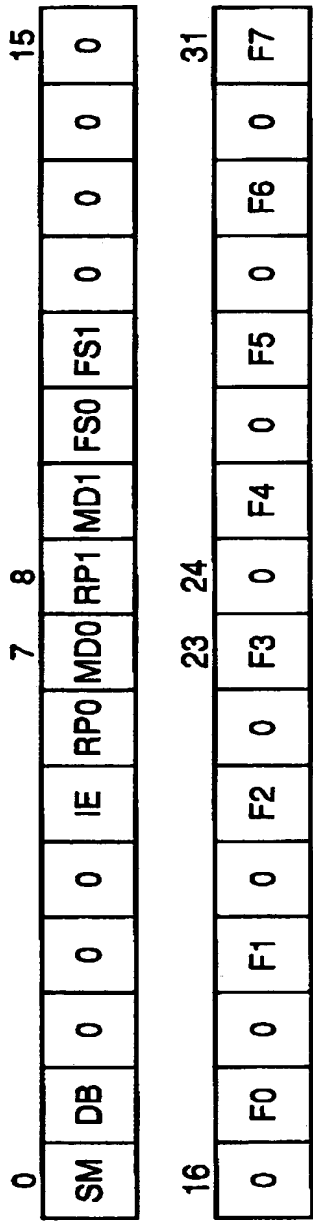
【図2】



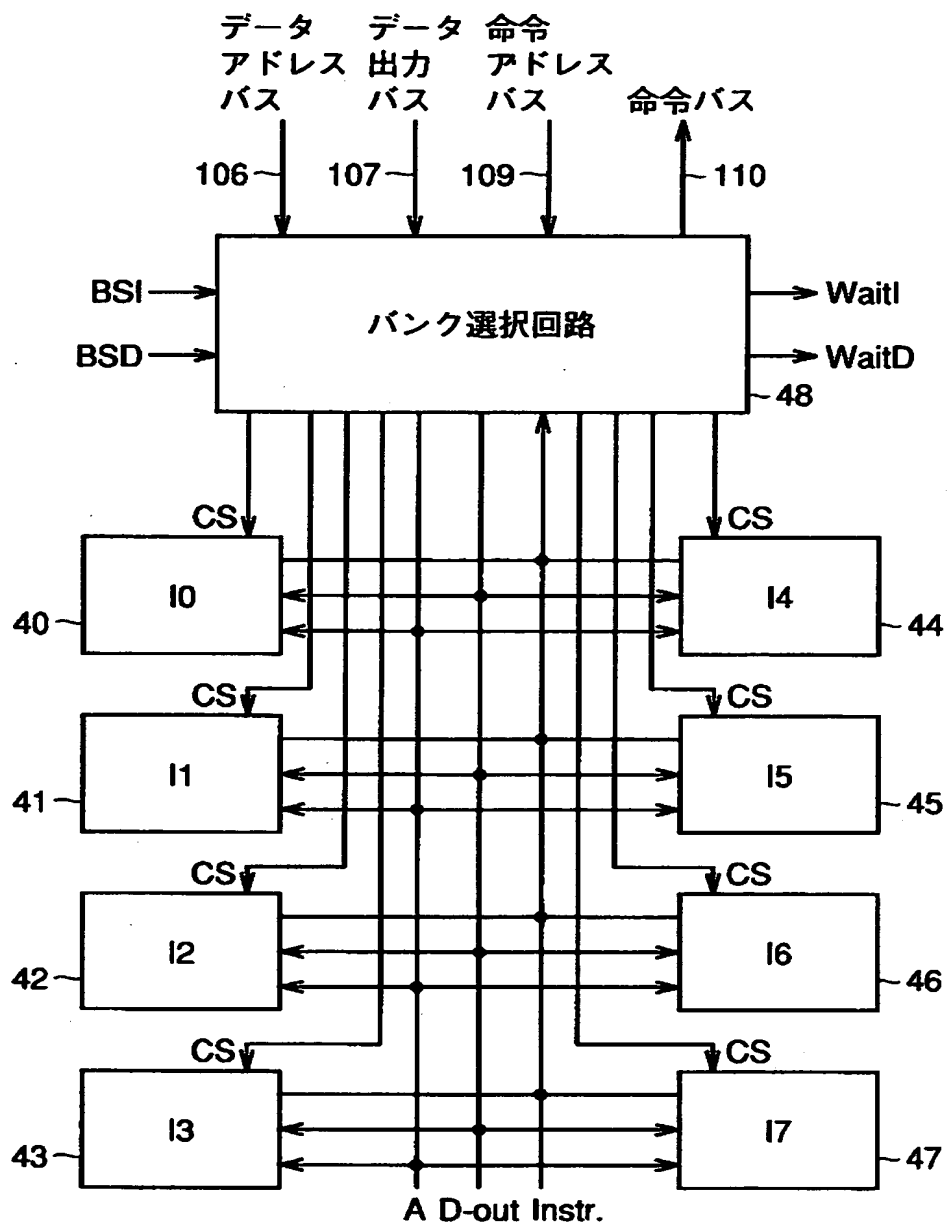
【図3】



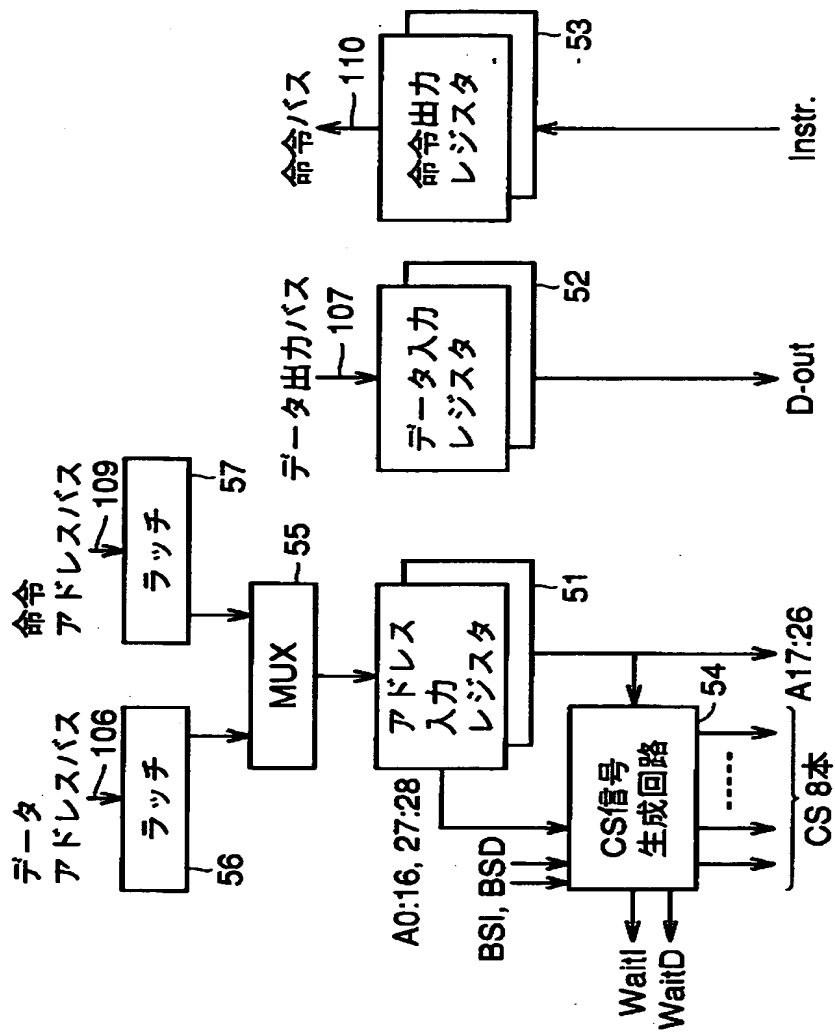
【図 4】



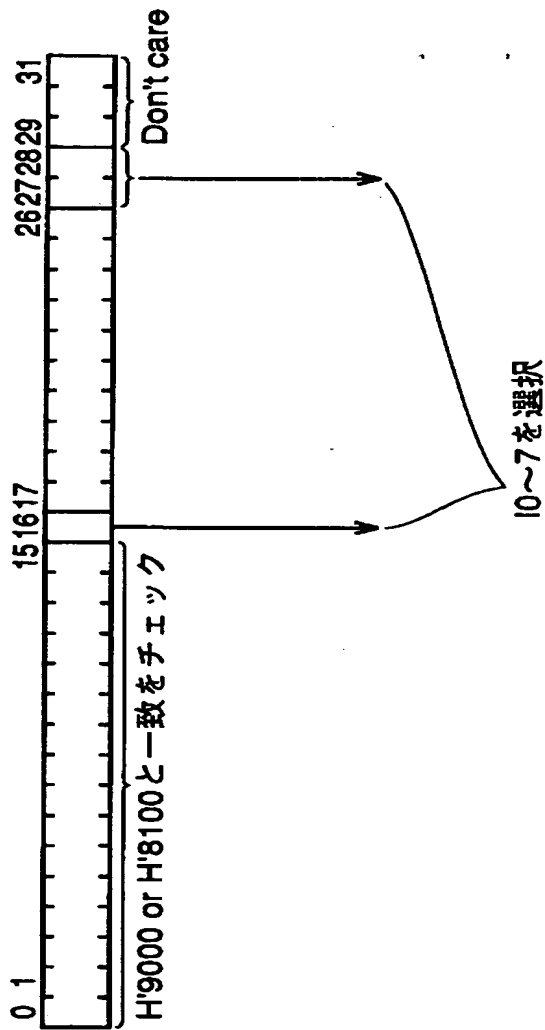
【図5】



【図6】



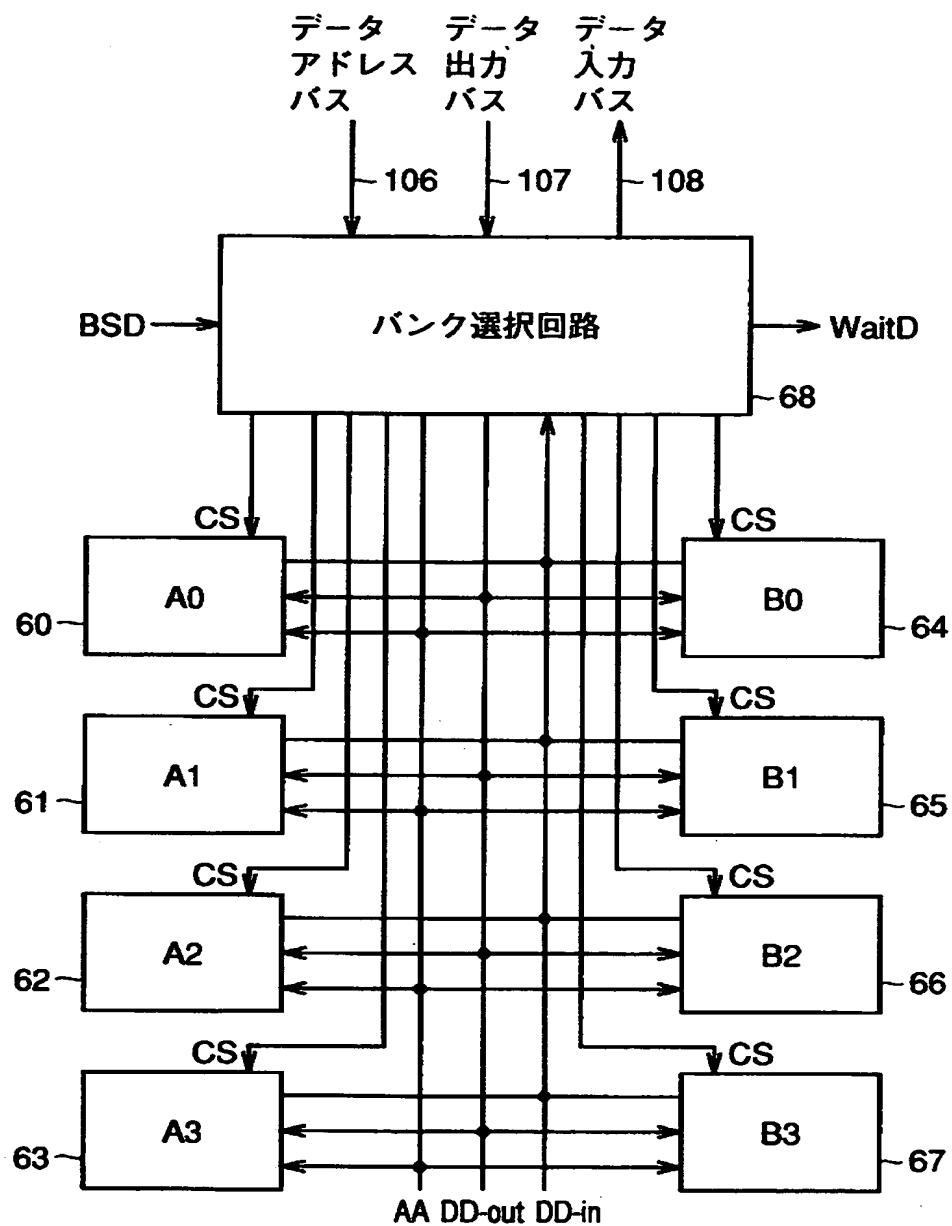
【図 7】



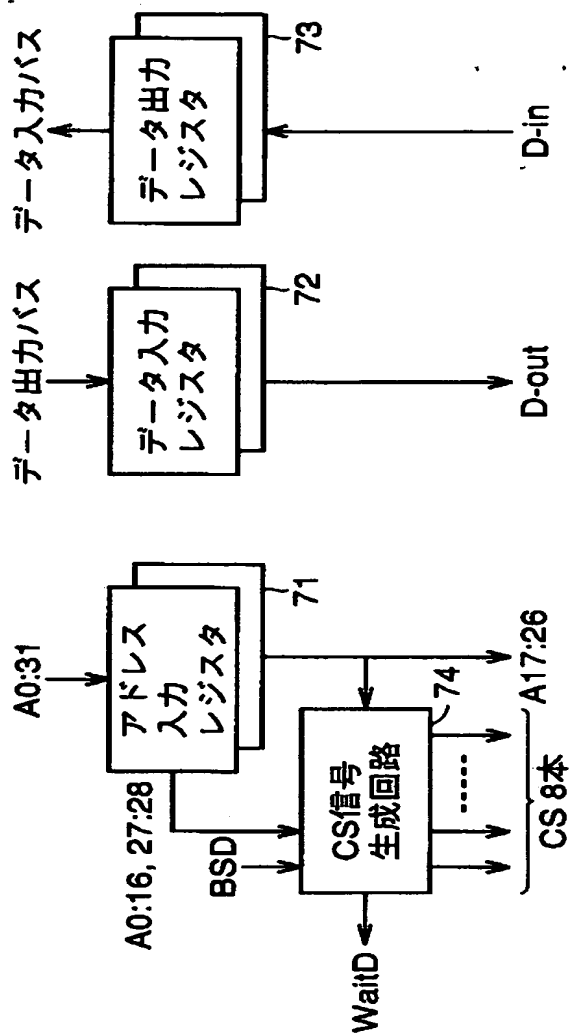
【図8】



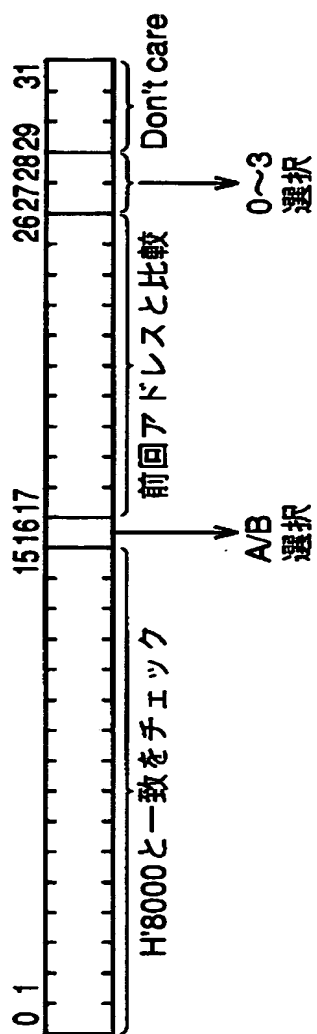
【図9】



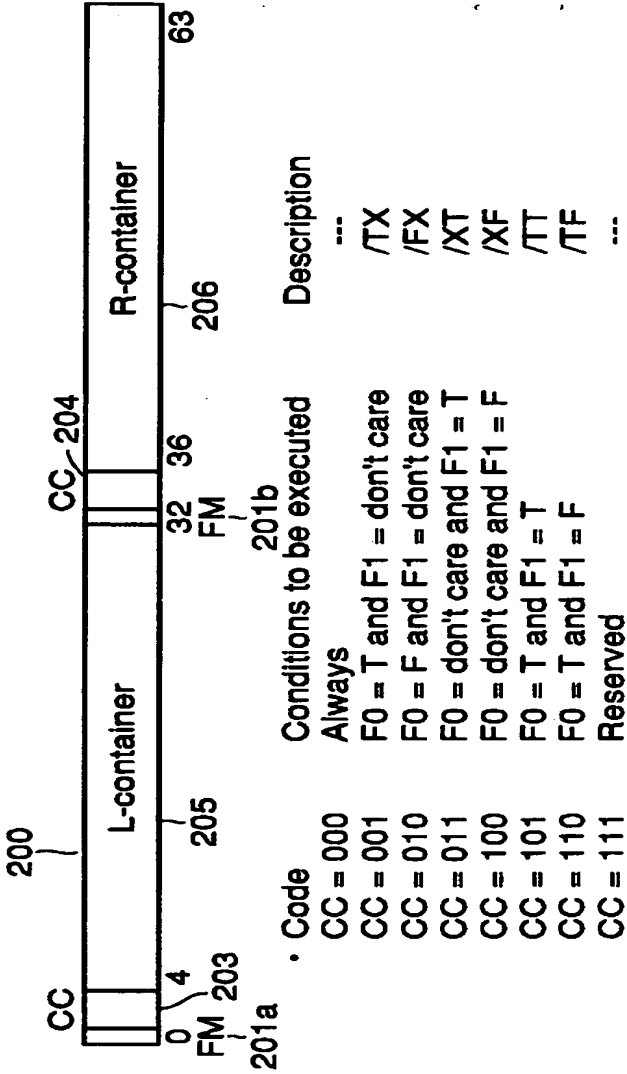
【図10】



【図11】

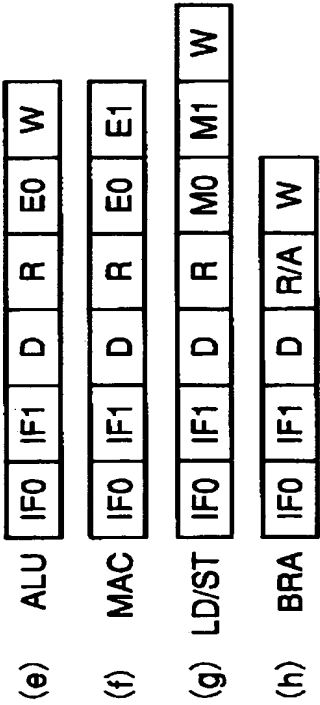
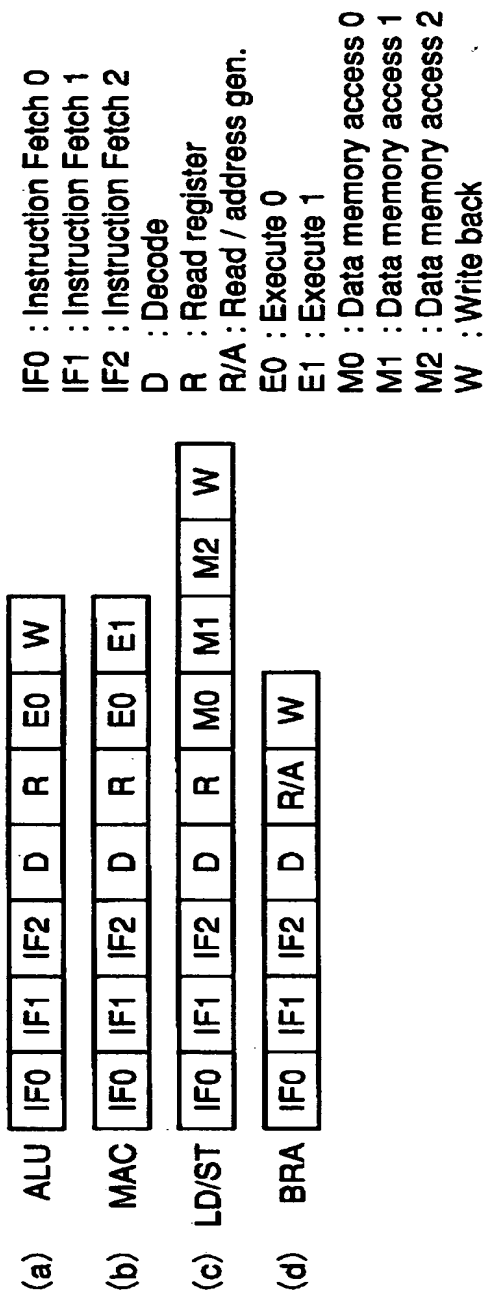


【図 1 2】

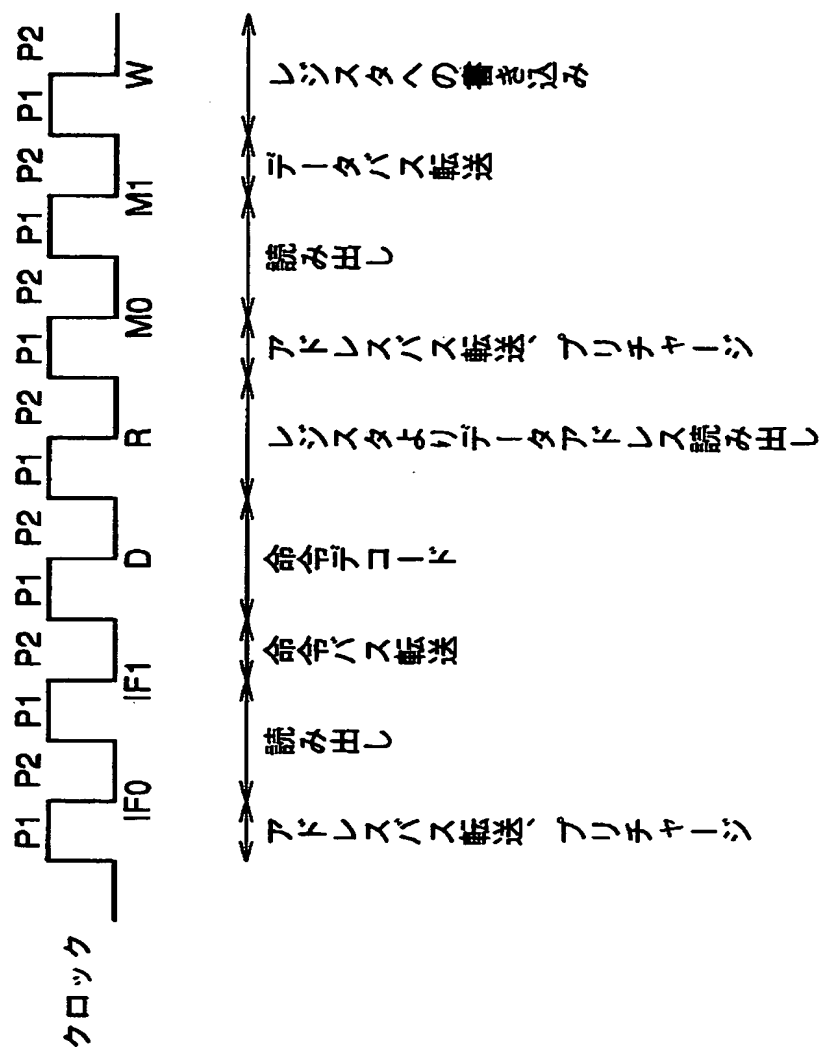


FM	Number of sub-instructions	Issuing Order	
		L-container	R-container
00	two	1st	1st
01	two	1st	2nd
10	two	2nd	1st
11	one	1st	----

【図 14】



【图 1·5】



【図 1. 6】

• Load/Store instructions

LDB	Load one byte to a register with sign extension
LDBU	Load one byte to a register with zero extension
LDH	Load one half-word to a register with sign extension
LDHH	Load one half-word to a register high
LDHU	Load one half-word to a register with zero extension
LDW	Load one word to a register
LD2W	Load two words to registers
LD4BH	Load four bytes to four half-word registers with sign extension
LD4BHU	Load four bytes to four half-word registers with zero extension
LD2H	Load two half-words to registers
STB	Store one byte from a register
STH	Store one half-word from a register
STHH	Store one half-word from a register high
STW	Store one word from a register
ST2W	Store two words from registers
ST4HB	Store four bytes from four half-word registers
ST2H	Store two half-words from registers
MODDEC	Decrement a register value by a 5-bit immediate value
MODINC	Increment a register value by a 5-bit immediate value

• Transfer instructions

MVFSYS	Move a control register to a general purpose register
MVTSYS	Move a general purpose register to a control register
MVFACC	Move a word from an accumulator
MVTACC	Move two general purpose registers to an accumulator

• Compare instructions

CMPcc	Compare cc = EQ (000), NE (001), GT (010), GE (011), LT (100), LE (101), PS - both positive (110), NG - both negative (111)
CMPUcc	Compare unsigned cc = GT (010), GE (011), LT (100), LE (101)

【図 1 7】

• Arithmetic operation instructions

ABS	Absolute
ADD	Add
ADDC	Add with carry
ADDHppp	Add half-word
	ppp = LLL (000), LLH (001), LHL (010), LHH (011), HLL (100), HLH (101), HHL (110), HHH (111)
ADDS	Add register Rb with the sign of the third operand
ADDS2H	Add sign to two half-word
ADD2H	Add two pairs of half-words
AVG	Average with rounding towards positive infinity
AVG2H	Average two pairs of half-words rounding towards positive infinity
JOINpp	Join two half-words
	pp = LL (00), LH (01), HL (10), HH (11)
SUB	Subtract
SUBB	Subtract with borrow
SUBHppp	Subtract half-word
	ppp = LLL (000), LLH (001), LHL (010), LHH (011), HLL (100), HLH (101), HHL (110), HHH (111)
SUB2H	Subtract two pairs of half-words

• Logical operation instructions

AND	logical AND
OR	logical OR
NOT	logical NOT
XOR	logical exclusive OR
ANDFG	logical AND flags
ORFG	logical OR flags
NOTFG	logical NOT a flag
XORFG	logical exclusive OR flags

• Shift operation instructions

SRA	Shift right arithmetic	
SRAHp	Shift right arithmetic a half-word	p = L (0), H (1)
SRA2H	Shift right arithmetic two half-words	
SRC	Shift right concatenated registers	
SRL	Shift right logical	
SRLHp	Shift right logical a half-word	p = L (0), H (1)
SRL2H	Shift right logical two half-words	
ROT	Rotate right	
ROT2H	Rotate right two half-words	

• Bit operation instructions

BCLR	Clear a bit
BNOT	Invert a bit
BSET	Set a bit
BTST	Test a bit

・【図 1 8】

・ Branch instructions

BRA	Branch
BRATZR	Branch if zero
BRATNZ	Branch if not zero
BSR	Branch to subroutine
BSRTZR	Branch to subroutine if zero
BSRTNZ	Branch to subroutine if not zero
DBRA	Delayed Branch
DBRAI	Delayed Branch immediate
DBSR	Delayed Branch to subroutine
DBSRI	Delayed Branch immediate to subroutine
DJMP	Delayed Jump
DJMPI	Delayed Jump immediate
DJSR	Delayed Jump to subroutine
DJSRI	Delayed Jump immediate to subroutine
JMP	Jump
JMPTZR	Jump if zero
JMPTNZ	Jump if not zero
JSR	Jump to subroutine
JSRTZR	Jump to subroutine if zero
JSRTNZ	Jump to subroutine if not zero
NOP	No operation

・ OS-related instructions

TRAP	Trap
REIT	Return from exception, interrupts, and traps

・ DSP Arithmetic operation instructions

MUL	Multiply	
MULX	Multiply with extended precision	
MULXS	Multiply and shift to the left by one with extended precision	
MULX2H	Multiply two pairs of half-words with extended precision	
MULHXpp	Multiply two half-words with extended precision	
	pp = LL (00), LH (01), HL (10), HH (11)	
MUL2H	Multiply two pairs of half-words	
MACd	Multiply and add	(d = 0, 1)
MACSd	Multiply, shift to the left by one, and add	(d = 0, 1)
MSUBd	Multiply and subtract	(d = 0, 1)
MSUBSd	Multiply, shift to the left by one, and subtract	(d = 0, 1)
SAT	Saturate	
SATHH	Saturate word operand into high half-word	
SATHL	Saturate word operand into low half-word	
SATZ	Saturate into positive number	
SATZ2H	Saturate two half-words into positive number	
SAT2H	Saturate two half-word operands	

・ Repeat instructions

REPEAT0	Repeat a block of instructions #0
REPEAT1	Repeat a block of instructions #1

・ Debugger supporting instructions

DBT	Debug trap
RTD	Return from debug interrupt and trap

【図 1 9】

H'0000 0000	← 8 バイト →
	なし
H'2000 0000	ROM21
H'4000 0000	SDRAM22
H'7000 0000	その他の プロセッサ10
H'7FFF FFF8	外部
H'8000 0000	低電力 データメモリ104
H'8100 0000	低電力 命令メモリ103
H'8200 0000	なし
H'8400 0000	高速 データメモリ102
H'8401 0000	なし
H'8402 0000	高速 命令メモリ101
H'8403 0000	その他

【図 2.0】

●Operation:

```

REPEAT1 #count, #pcaddr
  RPT1_C = #count-1
  RPT1_S = PC + 8
  RPT1_E = PC + pcaddr
  RPT1_I(0:5) = Instructions at memory((PC+8):(PC+48))
  if (PC==RPT1_E && RPT1_C > 0) {
    RPT1_C--
    PC == RPT1_S
  }

```

●Example:

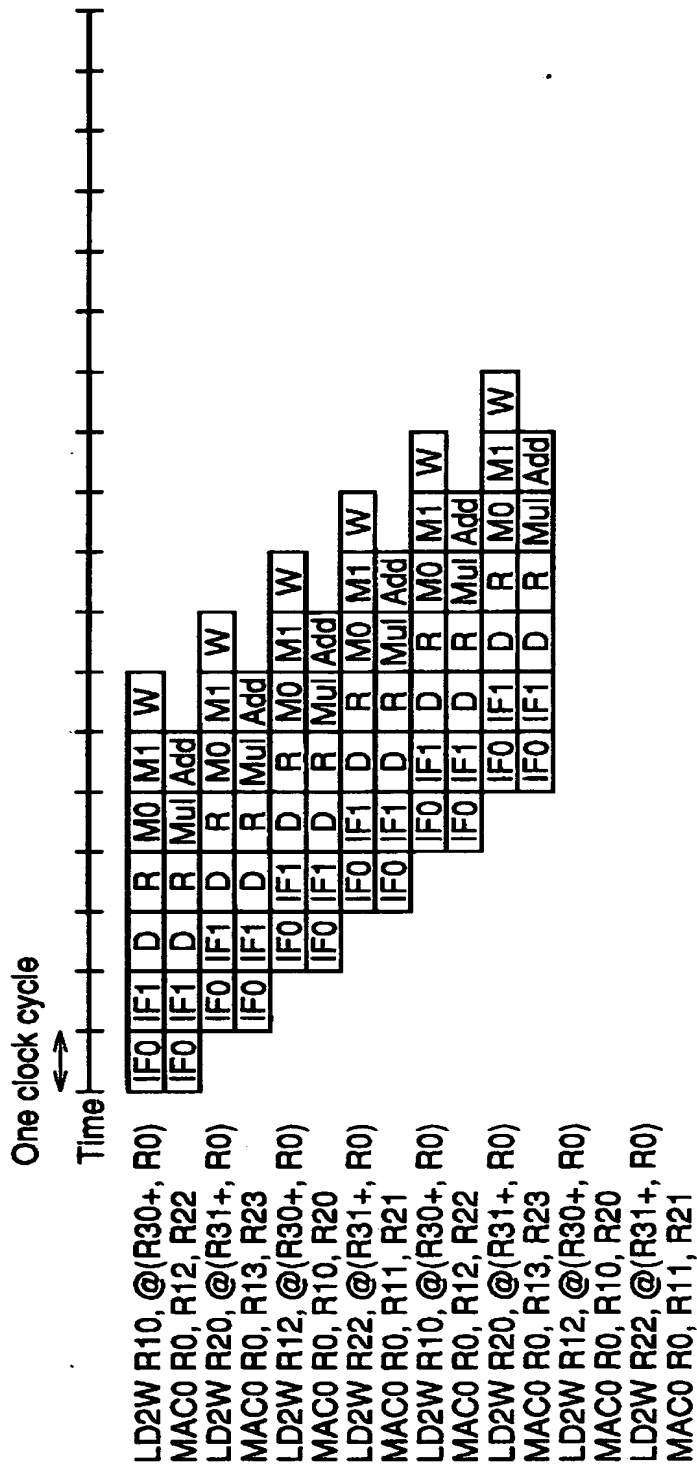
```

REPEAT1 #20, #48
START:LD2W R10, @(R30+, R0) || MAC0 R0, R12, R22
      LD2W R20, @(R31+, R0) || MAC0 R0, R13, R23
      LD2W R12, @(R30+, R0) || MAC0 R0, R14, R24
      LD2W R22, @(R31+, R0) || MAC0 R0, R15, R25
      LD2W R14, @(R30+, R0) || MAC0 R0, R16, R26
      END:LD2W R24, @(R31+, R0) || MAC0 R0, R17, R27

```

Zero-overhead loop

【図 2' 1】



【図 2 2】

●Operation:

```

REPEAT0 #count, #pcaddr
  RPT0_C = #count-1
  RPT0_S = PC + 8
  RPT0_E = PC + pcaddr
  RPT0_I = Instruction at memory(PC+8)
  if (PC == RPT0_E && RPT0_C > 0) {
    RPT0_C--
    PC == RPT0_S
  }

```

●Example:

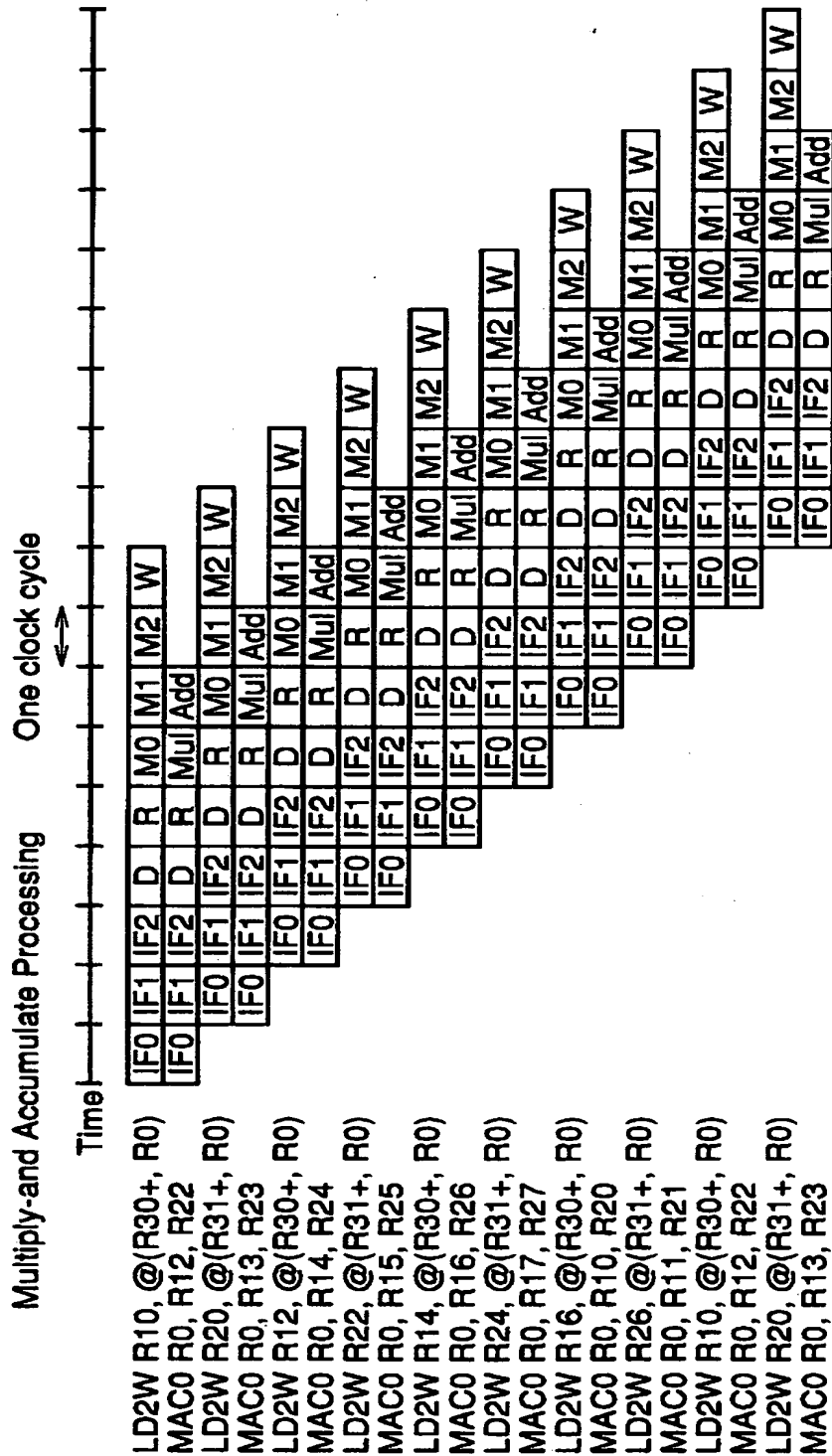
```

      REPEAT0 #10, #64
START:LD2W R10, @(R30+, R0) || MAC0 R0, R12, R22
      LD2W R20, @(R31+, R0) || MAC0 R0, R13, R23
      LD2W R12, @(R30+, R0) || MAC0 R0, R14, R24
      LD2W R22, @(R31+, R0) || MAC0 R0, R15, R25
      LD2W R14, @(R30+, R0) || MAC0 R0, R16, R26
      LD2W R24, @(R31+, R0) || MAC0 R0, R17, R27
      LD2W R16, @(R30+, R0) || MAC0 R0, R10, R20
END:LD2W R26, @(R31+, R0) || MAC0 R0, R11, R21

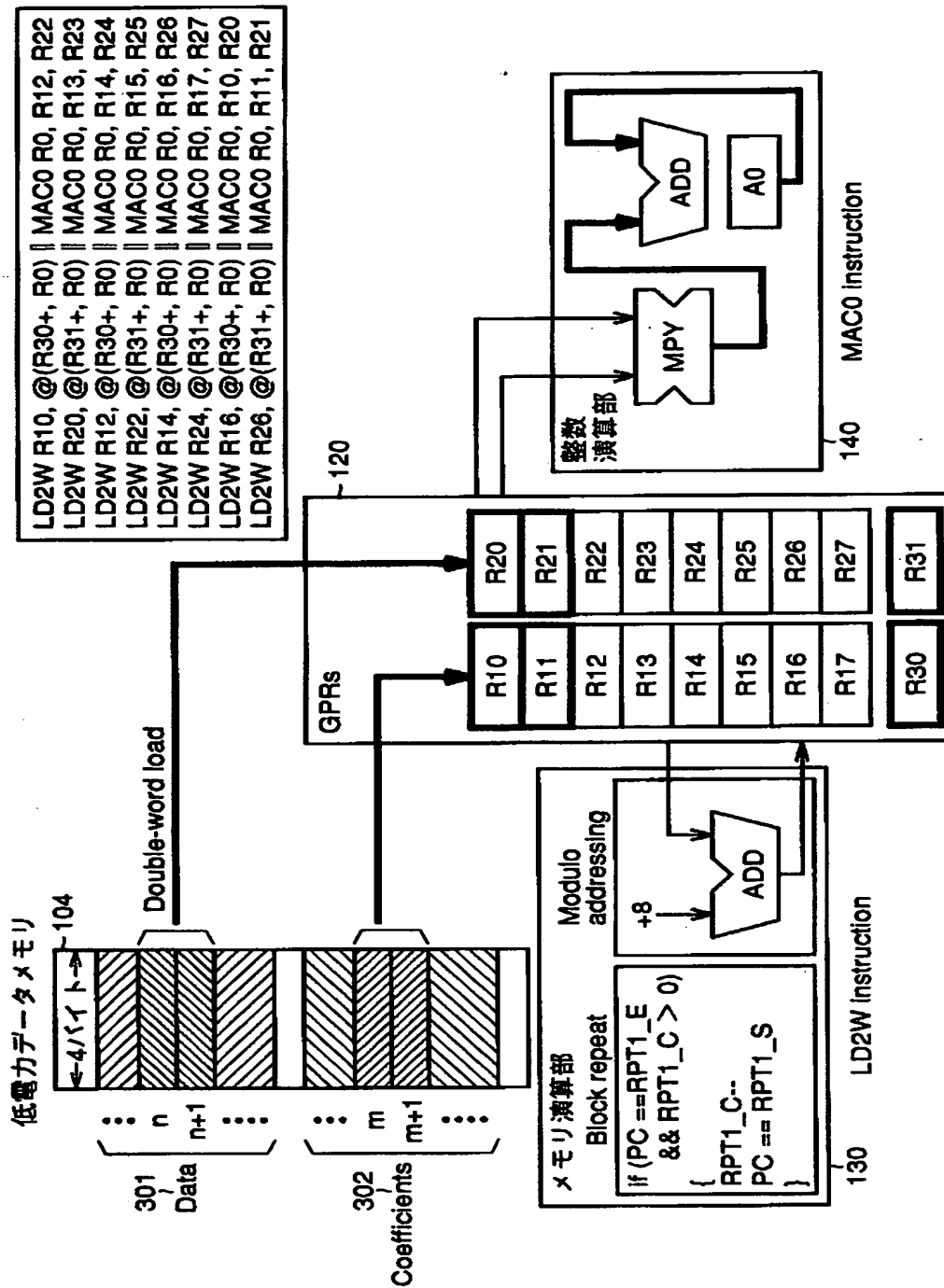
```

Zero-overhead loop

【図 23】



..【図2'4】



【書類名】 要約書

【要約】

【課題】 低速で低消費電力のメモリを用いつつも高速に処理することができ、処理性能の向上を図ることが可能なデータ処理装置を提供すること。

【解決手段】 複数のメモリバンクから命令をフェッチする場合に、メモリバンクの選択に対応したパイプラインステージ I F 0 と、命令の読出しに対応したパイプラインステージ I F 1 とを発生させてパイプライン処理を行なう。したがって、選択したメモリバンクのみをプリチャージすることができ、消費電力を削減することが可能となる。また、パイプラインステージ I F 0 と I F 1 とが並列に行なわれるため、命令メモリのスループットを向上させることが可能となる。

【選択図】 図 8

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 6 0 1 3]

1. 変更年月日	1 9 9 0 年 8 月 2 4 日
[変更理由]	新規登録
住 所	東京都千代田区丸の内 2 丁目 2 番 3 号
氏 名	三菱電機株式会社